Panasonic

BASIC入門

- BASIC編
- MSX-DOS2編
- 資料編



MSX turbo R

Panasonic
MSX turbo R
BASIC入門

この本でBASICやMSX-DOSを勉強されるかたへ

この本は、初めてパーソナルコンピュータ（以降、「パソコン」）に触れるかたのために、ＭＳＸ turbo Ｒパソコンに搭載されているプログラミング言語「漢字ＢＡＳＩＣ」とＭＳＸ turbo Ｒの標準オペレーティングシステム「ＭＳＸ－ＤＯＳ２」の基本的なことを説明しています。

この本をお読みになるかたに、次のことを守ってくださるようお願いします。

・先に、パソコンに付属している「取扱説明書」をお読みください。
・前のページから順番にお読みください。
・できるだけ、説明のとおりにパソコンを操作しながらお読みください。
・途中でわからなくなったら、前のほうの説明から読み直してみてください。

※この本では一部の命令や関数についての説明を省略しています。
この本に書いてあることがらがひと通り分かるようになりましたら、書店やパソコンの販売店などで販売されているＭＳＸパソコン用のＢＡＳＩＣの解説書などをお読みになるようお勧めします。(218ページに「推奨図書一覧」を掲載しています。)

表記上の約束

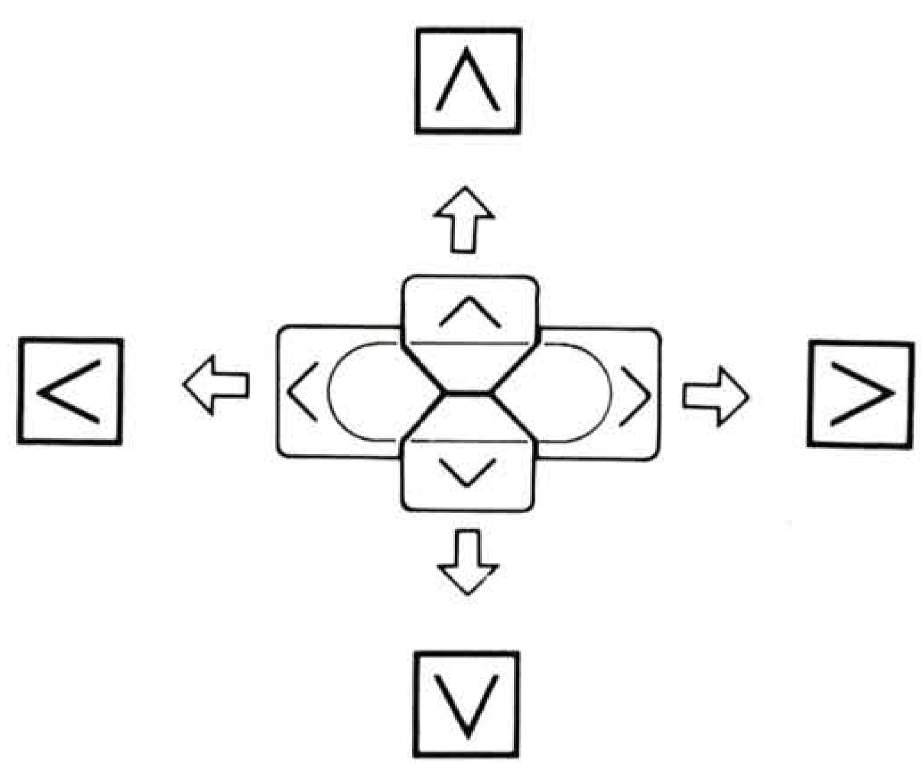
この本は次のような約束にしたがって書かれています。最初に覚えてください。

●説明の中の記号の意味は

ご注意	特に注意して欲しいことを説明します。 必ず読んでください。
◆	特定の場合にだけあてはまる説明をします。 ご自分にあてはまるところだけ読んでください。
メモ ※	各項目の補足説明や、知っておくと便利なことがらを説明しています。
（⇨●●ページ）	参照ページを示します。

●キーの表記（書き表しかた）は

・カーソルキーは次のように表記します。



・必要な文字だけを表示する場合は、その文字だけを［　］で囲んで表記します。



・表の中などでは、あるキーを押しながら別のキーを押すことを、２つのキーを＋記号でつないで表記する場合があります。

例）　［CTRL］＋［STOP］

［CTRL］キーを押しながら［STOP］キーを押す意味です。

・［　］は［SPACE］をあらわします。

●下のようなマークは

💾 ････ フロッピーディスクドライブ　　漢 ････ 漢字プリンタ
🖨 ････ プリンタ

その部分の説明のとおりに操作を行うのに必要な機器をあらわします。

BASIC編

第1章 初めてのプログラム

第2章 体験! プログラミング

第3章 お待たせ! グラフィック

第4章 BASICでMUSIC!

第5章 かんたん漢字モード!

第6章 サンプルプログラム集

MSX-DOS2編

資　料　編

もくじ

パソコンって何だろう？ …… 8
「ＢＡＳＩＣ」って何だろう？ …… 9

ＢＡＳＩＣ編

第１章　初めてのプログラム

ＢＡＳＩＣのスタート …… 12
ＢＡＳＩＣの終了のしかた …… 13
キーボードを使いこなそう …… 14
ＢＡＳＩＣの２つの顔（ダイレクトモードとプログラムモード）…… 16
パソコンに直接命令してみよう（ダイレクトモード）…… 20
プログラムに挑戦しよう …… 24
プログラムを入力しよう …… 28
プログラムの実行　――失敗しても当たり前 …… 30
プログラムの保存と呼び出し …… 34

第２章　体験！プログラミング

ＢＡＳＩＣの約束ごと …… 44
白いヨーヨー …… 46
数当てゲーム …… 50
文字も数値で表せる？ …… 54
謎の三角形 …… 56
記憶力ゲーム …… 58

第３章　お待たせ！グラフィック

グラフィック画面へ …… 62
直線、四角、円 …… 66
いろいろな色 …… 68
スクリーン８は宝石箱 …… 70
レモンを描こう …… 74
スプライトって何だろう …… 76

第４章　ＢＡＳＩＣでＭＵＳＩＣ！
まず♪ドレミから …… 84
ＦＭ音源で演奏してみよう …… 88
キーボードで音楽を …… 92
ＰＣＭ音源で録音・再生してみよう …… 95

第５章　かんたん！漢字モード
漢字モードのスタート …… 98
漢字の入力のしかた …… 100
漢字モードの種類 …… 111
漢字モードの画面 …… 112
漢字で住所氏名を表示しよう …… 114
漢字グラフィック画面で文字表示 …… 116
「入門」のおわりに …… 118

第６章　サンプルプログラム集
ＵＦＯ …… 120
ヘリコプター …… 123
お絵かきプログラム …… 127
住所録作成プログラム …… 131

ＭＳＸ－ＤＯＳ２編

第１章　「ＭＳＸ－ＤＯＳ２」の紹介
「ＭＳＸ－ＤＯＳ２」って何だろう？ …… 136

第２章　ＭＳＸ－ＤＯＳ２の使いかた
ＭＳＸ－ＤＯＳ２を使ってみよう …… 140

資　料　編

漢字ＢＡＳＩＣの規則概説 …… 152
キャラクタコード表 …… 174
ＢＡＳＩＣリファレンス表（ＤＩＳＫ　ＢＡＳＩＣ２を含む） …… 175
ＢＡＳＩＣ予約語表 …… 194
コントロールコード表 …… 195
メモリマップ（ＢＡＳＩＣ使用時） …… 196
メモリマップ（ＭＳＸ－ＤＯＳ２使用時） …… 198
ＭＳＸ－ＤＯＳ２リファレンス表 …… 199
ＢＡＳＩＣエラーメッセージ表 …… 207
ＭＳＸ－ＤＯＳ２エラーメッセージ表 …… 209
ローマ字・かな変換表（ＢＡＳＩＣ使用時） …… 212
命令索引 …… 214
推奨図書一覧 …… 218
漢字コード表の補足説明 …… 219

パソコンって何だろう?

パソコンを使い始める前に、まずパソコンとはどんなものなのか、他の電気製品とはどんなところが違うのかを理解しておきましょう。

パソコンの特徴は

パソコンとテレビや冷蔵庫のような普通の電気製品とのあいだには、次のような違いがあります。

1．使いみちを決めるのは

テレビや冷蔵庫の使いみちはだれでも知っています。まさかテレビでスイカを冷やそうとしたり、冷蔵庫で「今日のニュース」を見ようとする人はいないでしょう。

これらの「電気製品」は、工場で作られたときから使いみちが決められているのです。

パソコンは少し違います。工場では使いみちを決めないで作られていますし、お店の人も「これは○○に使うものだ」と決めて売ったりはしていません。

パソコンを何に使うのかは、あなた自身が決めることなのです。

2．目的によって使い分けるには

テレビや冷蔵庫なら、最初に正しく接続してしまえば、あとはスイッチを入れるだけで働き始めます。パソコンは違います。接続して電源スイッチを入れるだけでは何もしてくれません。どんな仕事をしたらよいのかをまだパソコンに指示していないからです。

こういうと、「いや、そんなことはない。ぼくのパソコンはスイッチを入れるだけでテレビにきれいな画面が映るよ」という人もいるでしょう。たしかにパソコンの機種によっては、そのような動きかたをするものもあります。でもそれは、電源が入ったときにはそういう仕事をしろという指示が、あらかじめ工場でパソコンにあたえられているからです。

パソコン本体やそれに接続して使う各種の機器は、目で見ることや手で触ることができます。これに対してパソコンに仕事をさせる指示は、見ることも触ることもできません。しかし、パソコンに何かの働きをさせるには、どちらも同じように大切なものなのです。

そこで、パソコン本体やその他の機器のことを**「ハードウェア」**（**「ハード」**）と呼び、パソコンにあたえる指示などのことを**「ソフトウェア」**（**「ソフト」**）と呼んで対等に扱います。

パソコンが実際にどんな働きかたをするかは、どんなソフトをあたえるかによって決まるのです。

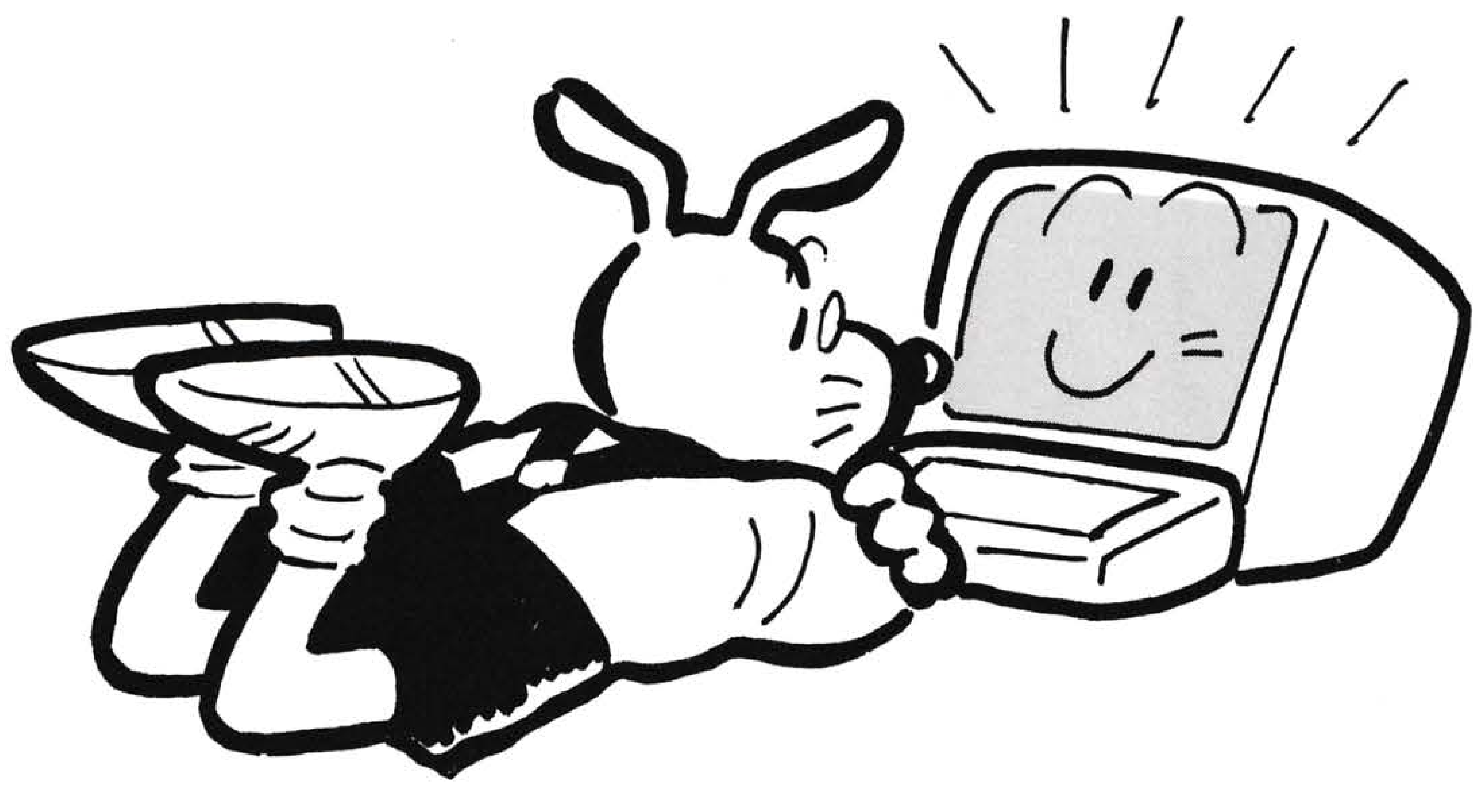
「BASIC」って何だろう?

この本の表紙にもある「ＢＡＳＩＣ」(ベーシック)とは、いったい何でしょうか? また、「ＭＳＸ turbo Ｒ」(エムエスエックス・ターボ・アール)とは何をあらわすのでしょうか? ここではその疑問に答えます。

「BASIC」とは

パソコンを動かすソフトはいろいろな方法で手に入れることができます。ゲームやワープロ(ワードプロセッサ)としてお店で売られているソフトもありますし、雑誌に掲載されることもあります。また、ソフトを自分で作ることもできます。

ソフトを自分で作るということは、パソコンに自分で命令をあたえるということです。でも、パソコンはただの機械ですから、人間のことばは分かりません。
そこで、機械であるパソコンと人間の両方に分かることばが必要になります。
人間がそのことばでパソコンに命令すれば、パソコンはあたえられた命令どおりに働くわけです。
このようなことばを**「プログラミング言語」**と呼びます。

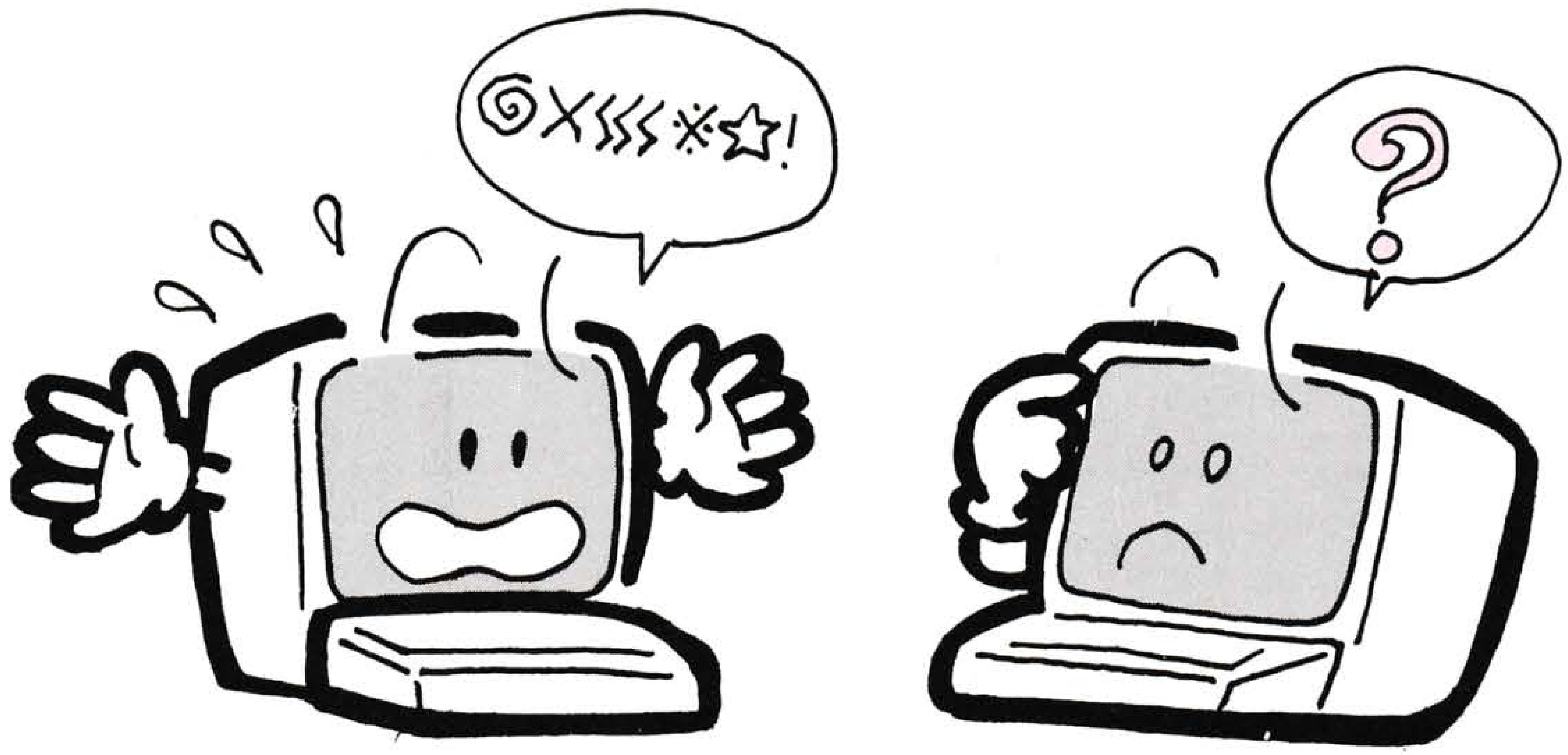


ＢＡＳＩＣはそのプログラミング言語のひとつです。
ＢＡＳＩＣの他にもたくさんのプログラミング言語が作られましたが、初めてプログラミング言語を学ぶ人にとっては、ＢＡＳＩＣが一番覚えやすくて親切なことばであるため、ほとんどのパソコンはＢＡＳＩＣで命令できるようになっています。

「MSX turbo R」とは

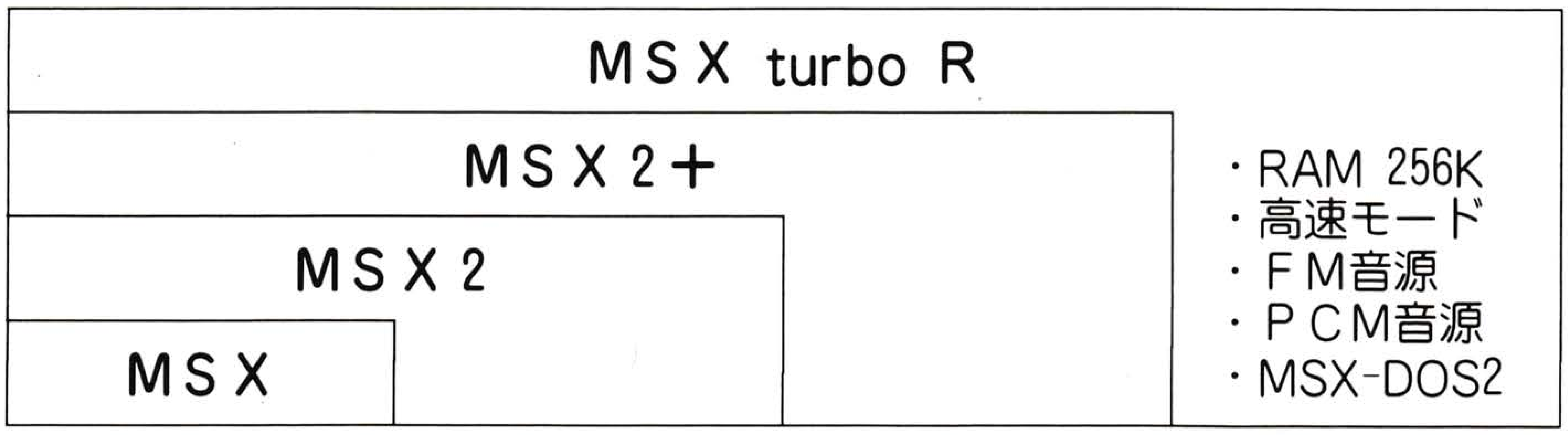
では、いろいろなパソコンのＢＡＳＩＣで作ったソフトを、おたがいに交換して使うことはできないのでしょうか？

結論から先に言うと、ほとんどの場合は、そのままでは使えません。なぜなら、パソコンの機種によって、ＢＡＳＩＣの命令の種類や文法（ことばの決まり）に少し違いがあるからです。



あなたのパソコンは「ＭＳＸ turbo Ｒパソコン」と呼ばれるものです。ＭＳＸ turbo Ｒパソコンにもいくつかの機種がありますが､すべて「ＭＳＸ turbo Ｒ」という規格（共通の約束ごと）にしたがって作られています。ですから､ＭＳＸ turbo Ｒパソコンどうしなら、ＢＡＳＩＣで作ったソフトをお互いに交換して使うことができます。

ＭＳＸ turbo Ｒ規格は、その前に作られた「ＭＳＸ２＋」､「ＭＳＸ２」､「ＭＳＸ」という３つの規格をその中に含んでいます。



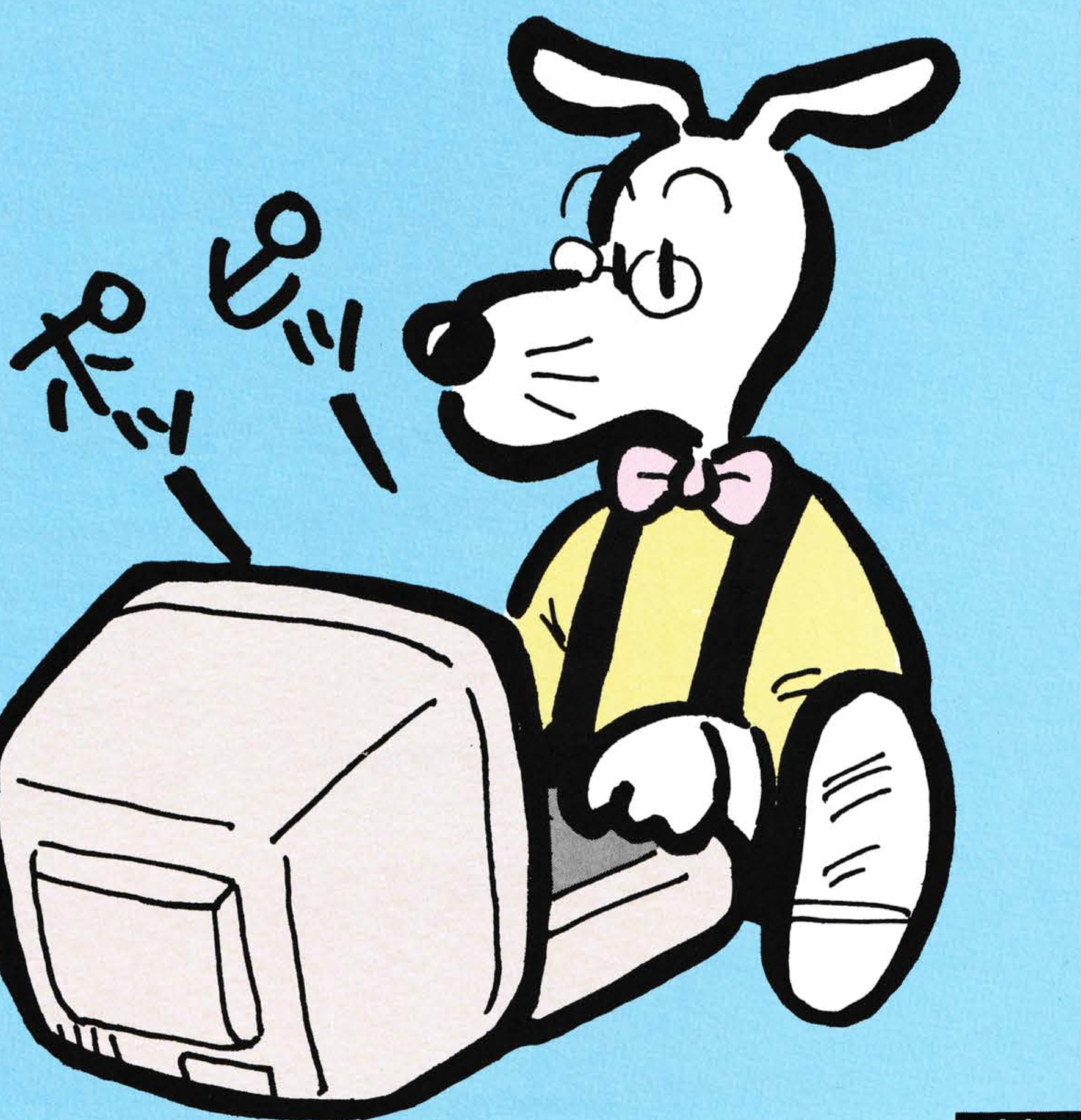
ですから、ＭＳＸ turbo Ｒパソコンでは、ＭＳＸパソコンやＭＳＸ２、ＭＳＸ２＋パソコンのために作られたソフトも使うことができます。
この本では､ＭＳＸ turbo Ｒパソコン用のＢＡＳＩＣである「漢字ＢＡＳＩＣ」とＭＳＸ－ＤＯＳ２の基本的な使いかたを説明しています。

第1章

プログラムを自分で作れば、今日からあなたもパソコンのエキスパートです。

初めてのプログラム

ＢＡＳＩＣのスタート ……………………………12
ＢＡＳＩＣの終了のしかた ………………………13
キーボードを使いこなそう ………………………14
ＢＡＳＩＣの２つの顔 ……………………………16
（ダイレクトモードとプログラムモード）
パソコンに直接命令してみよう（ダイレクトモード）……20
プログラムに挑戦しよう …………………………24
プログラムを入力しよう …………………………28
プログラムの実行――失敗しても当たり前 …………30
プログラムの保存と呼び出し………………………34



BASICのスタート

ＢＡＳＩＣのスタートのしかたは、パソコンの機種により異なります。
パソコンの取扱説明書を読んで確かめてください。

◆パソコンの機種によっては、電源スイッチを入れる前に「内蔵ソフト切替スイッチ」などを操作することが必要です。

接続とスタートが正しくできたら、しばらくしてテレビの画面に下の文字が表示されます。

```
MSX BASIC version 4.0
Copyright 1990 by Microsoft
XXXXX Bytes free
Disk BASIC version 2.XX
Ok
█
```

（高速モードの画面）

この数字が「２」なら高速モード、「１」なら標準モードの画面です。また、市販ソフトなどを使用した後で、電源スイッチを切らずにＢＡＳＩＣに戻ったときには、表示が異なる場合があります。

画面に「Ｏｋ」という文字が、その下に四角いマーク（カーソル）が出ていればＢＡＳＩＣが正常にスタートしています。

◆いつまでも上の表示が画面に出てこなかったり、何も画面に表示されないときは、接続のしかたかＢＡＳＩＣのスタートのしかたをまちがえていますので、一度スイッチを切り、パソコンの取扱説明書を読み直しながら、接続とＢＡＳＩＣのスタートをやり直してください。

◆ＭＳＸ turbo Ｒパソコンは、「高速モード」と「標準モード」の２種類のスピードで動作します。このモードにより、ＢＡＳＩＣの機能やプログラムの実行速度が変わります。この本では、高速モードで動作するときのＢＡＳＩＣ（「ＤＩＳＫ　ＢＡＳＩＣ２」）をもとに説明しています。標準モードで使えない命令について、必要なときは「ＢＡＳＩＣリファレンス表」で確認してください。
高速モードと標準モードの切り替えは、パソコンの電源スイッチを入れたときやリセットボタンを押したときに自動的に行われます。
通常、ＭＳＸ turbo Ｒパソコンは高速モードで動作します。ＭＳＸ、ＭＳＸ２、ＭＳＸ２＋パソコンで作成したＢＡＳＩＣのプログラムや、標準モードで作成したプログラムも、高速モードの画面を表示させてからプログラムを呼び出して実行させることにより、高速で動作させることができます。ただし、音などに時間的なずれが生じる場合があります。
以下の場合は、標準モードで動作します。

- **「ＭＳＸ turbo Ｒ」マークが表示されていないＭＳＸ規格の市販ソフト（ＲＯＭカートリッジやフロッピーディスク）をスタートしてから、電源スイッチを切らずにＢＡＳＩＣをスタートしたとき**
- **ＭＳＸ、ＭＳＸ２、ＭＳＸ２＋パソコンでフォーマット（初期化）したフロッピーディスク、またはＭＳＸ turbo Ｒパソコンが標準モードで動作しているときにフォーマットしたフロッピーを、ディスクドライブに入れた状態で次の操作をしたとき**
  - **・内蔵ソフト切替スイッチを「切」にして電源を入れた。**
  - **・内蔵ソフト切替スイッチを「切」にしてリセットボタンを押した。**
  - **・内蔵ソフトのメインメニュー画面でＢＡＳＩＣをスタートした。**

BASICの終了のしかた

ＢＡＳＩＣの終了のしかたは、カンタンです。
ふつうは、ただパソコンの電源スイッチを切るかリセットボタンを押せば、
それでＢＡＳＩＣは終了します。
ただし、次の場合はちょっと待ってください。

アクセスランプが点灯しているとき

フロッピーディスクドライブのアクセス（または ＩＮ ＵＳＥ）ランプが点灯しているときは、ランプが消えるまで終了しないでください。

プログラムを作ったとき

パソコンはポータブルのワープロなどとは異なり、電源スイッチを切ったりリセットボタンを押すと、入力したものがすべて消えてしまいます。消えては困る大切なプログラムなどを入力したときは、電源スイッチを切る前に保存しておきましょう。
プログラムの保存のしかたは、34ページから説明しています。

キーボードを使いこなそう

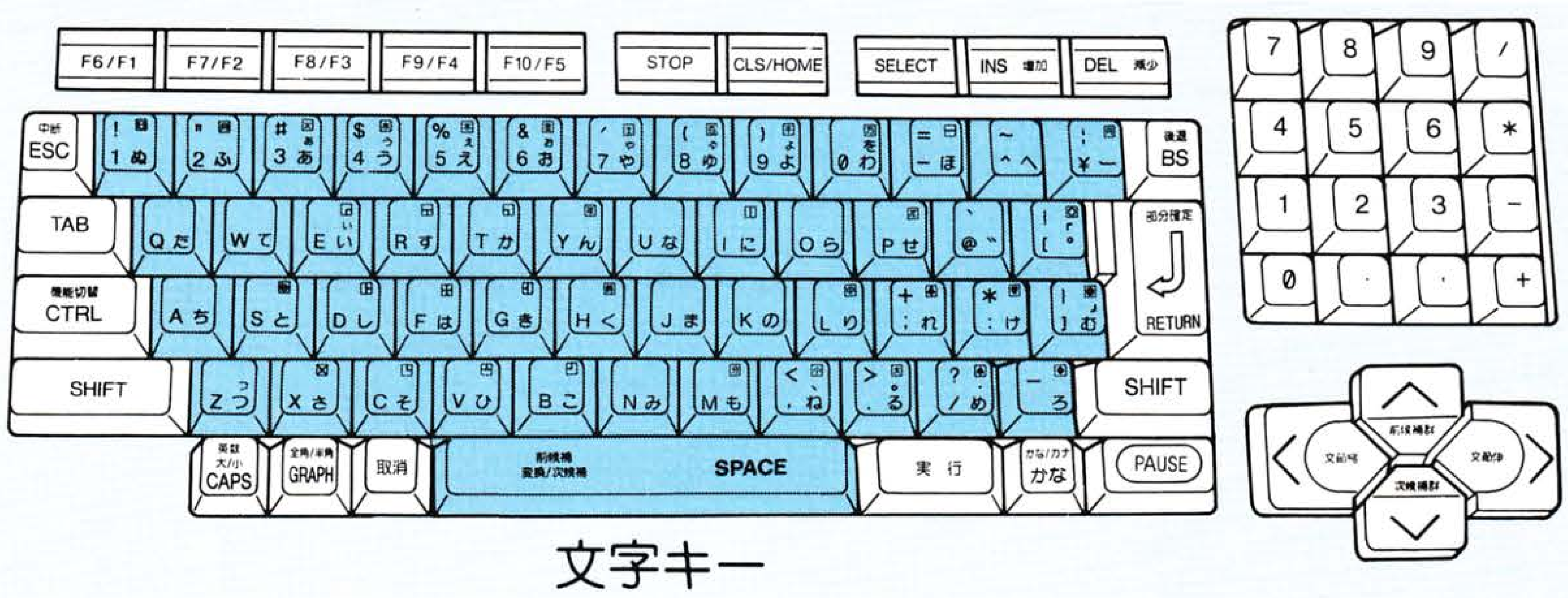
パソコンに命令をあたえるには、ふつうキーボードを使います。ですから、キーボードをうまく使えないうちは、パソコンをうまく使えないことになります。
まずキーボードの使いかたをひととおり知っておきましょう。

文字キーと特殊キー

キーボードに触れるのは初めてだという人は、たぶん「こんなにたくさんのスイッチのようなものを操作しないと使えないのなら、使いかたを覚えるのはたいへんだろうな」と思われたでしょう。

たしかにキーボードにはたくさんのキーが並んでいます。でも、そのほとんどは、キーの上に書かれている文字を一文字だけ画面に表示する働きしかしません。ですから、ただの鉛筆やボールペンのようなものだと思ってください。
これらのキーを**「文字キー」**と呼びます。

例）キーボード図



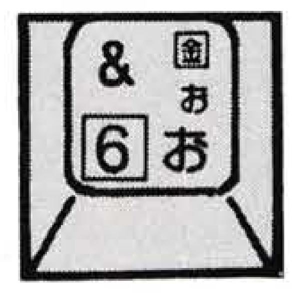
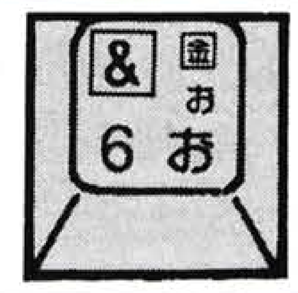
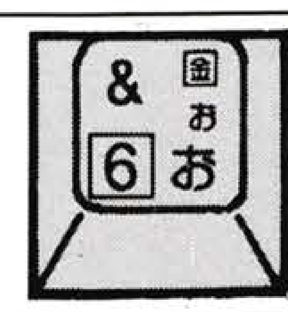
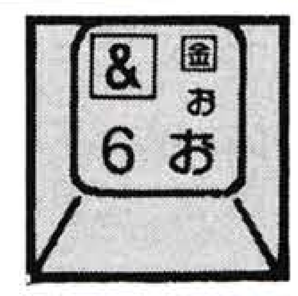
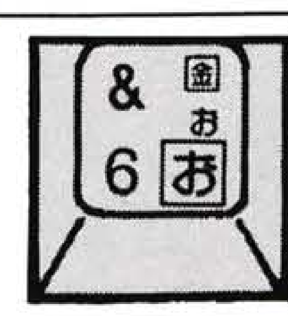
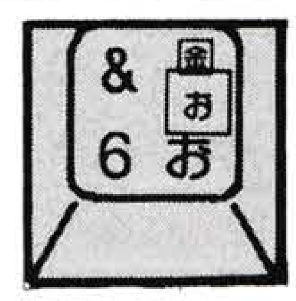
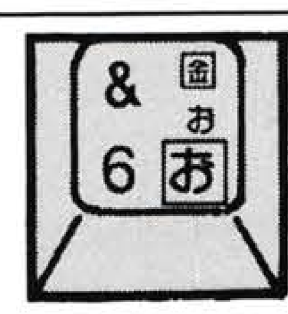
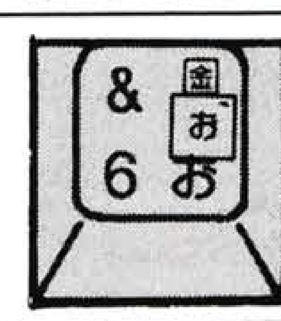
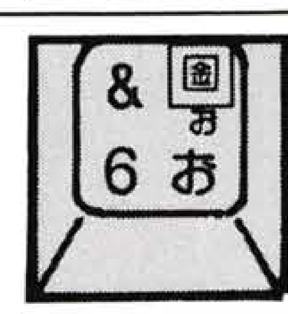
キーボードにある他のキーは、**「ファンクションキー」**とか**「特殊キー」**とか呼ばれるものでいろいろな機能をもっています。
その中で、下のキーだけは必ず覚えておいてください。

↵（リターン）キー	………………………	表示した文字の入力
△▽◁▷（カーソル）キー	………………………	表示位置の移動
SHIFT（シフト）キー	………………………	表示する文字の切替
かな キー	………………………	表示する文字の切替
GRAPH キー	………………………	表示する文字の切替

メモ：これ以降、特に理由がない限り、キーは「SHIFT」のように表記し、「キー」を省略します。

キーボードの使いかたの基本はパソコンの取扱説明書に書かれていますが、かんたんに整理すると、次ページの表のとおりです。

文字の表示のしかた

CAPS ランプ	かな ランプ	文字キーだけを押した時 表示される文字		SHIFT を押しながら文字キーを押した時 表示される文字	
消	消	英小文字、数字、英記号		英大文字、英記号	
点灯	消	英大文字、数字、英記号		英小文字、英記号	
消	点灯	ひらがな、かな記号		ひらがな、かな記号	
点灯	点灯	カタカナ、かな記号		カタカナ、かな記号	
GRAPH を押しながら		グラフィック文字			

※ CAPSランプ：CAPS を押すと点灯し、もう一度押すと消える。
かなランプ　：かな を押すと点灯し、もう一度押すと消える。

※ SHIFT を押しながら かな を押して ローマ字のつづりにしたがって英文字のキーを押すと、ひらがな（CAPSランプ消灯時）またはカタカナ（CAPSランプ点灯時）を画面に表示することができます。
（⇨212ページ）

BASICの2つの顔(ダイレクトモードとプログラムモード)

ＢＡＳＩＣの使いかたには、大きく分けて２種類あります。
ひとつは「ダイレクトモード」、もうひとつは「プログラムモード」です。
では、２つはどんなところが違うのでしょうか？

⏎ 一発！ すぐ実行──ダイレクトモード

スタートしたときには、ＢＡＳＩＣはダイレクトモードになっています。ダイレクトモードでは、キーボードを使って人間がパソコンに直接命令します。命令するたびにすぐに実行されるので、ちょうど人間どうしが会話をするように命令することができます。
では、ダイレクトモードでパソコンに命令してみましょう。

「ＣＬＳ」（クリア・スクリーン）は、画面の表示をすべて消し、カーソルを画面の左上のすみに戻す命令です。
まず、次のようにキーをひと文字ずつ押してみてください。

[C] [L] [S]

押したキーの文字は画面に表示されますが、それ以上何も起こりません。
これは、パソコンがまだ命令を受け取っていないからです。



パソコンに命令を受け取らせるためには、文字キーを押して命令を画面に表示させたあと、⏎ を押すことが必要です。

このとき、カーソルは表示された命令と同じ行になければいけません。
このように、命令に続けて ⏎ を押し、パソコンに命令を受け取らせることを「命令を**入力する**」と呼びます。また、入力された命令にしたがって、パソコンが何かをすることを「命令を**実行する**」と呼びます。

では ⏎ を押してみましょう。画面は右のようになります。
これがＣＬＳ命令の実行を終わったときの状態です。

```
Ok
■




color auto  goto  list  run
```

命令の実行を終わると、パソコンはカーソルの位置に**「Ｏｋ」**と表示し、カーソルを次の行の左端に動かして次の命令を待ちます。

※ＢＡＳＩＣの命令は、小文字で入力しても同じように実行されます。

◆画面が上のようにならずに下の文字（シンタックス・エラー）が表示されたら、キーを押しまちがえていますから、もう一度正しく入力し直してください。

```
Syntax error
Ok
■
```

メモ：ＣＬＳ命令は、あとで説明する漢字テキスト画面では使えません。
漢字テキスト画面での画面の消去には、ＣＡＬＬ　ＣＬＳという命令を使用します。

まとめて実行——プログラムモード

ダイレクトモードでは、命令をひとつずつ入力して実行させます。しかし、パソコンに何かまとまった仕事をさせるときは、いくつもの命令が必要なので、ひとつずつ命令を入力しては実行させていたのではとてもたいへんです。

そこで、パソコンに実行させたい命令を順番に並べた**「プログラム」**というものを作ってあらかじめパソコンに入力し、入力が終わってからプログラムを実行させる命令を入力します。

これが**「プログラム モード」**です。

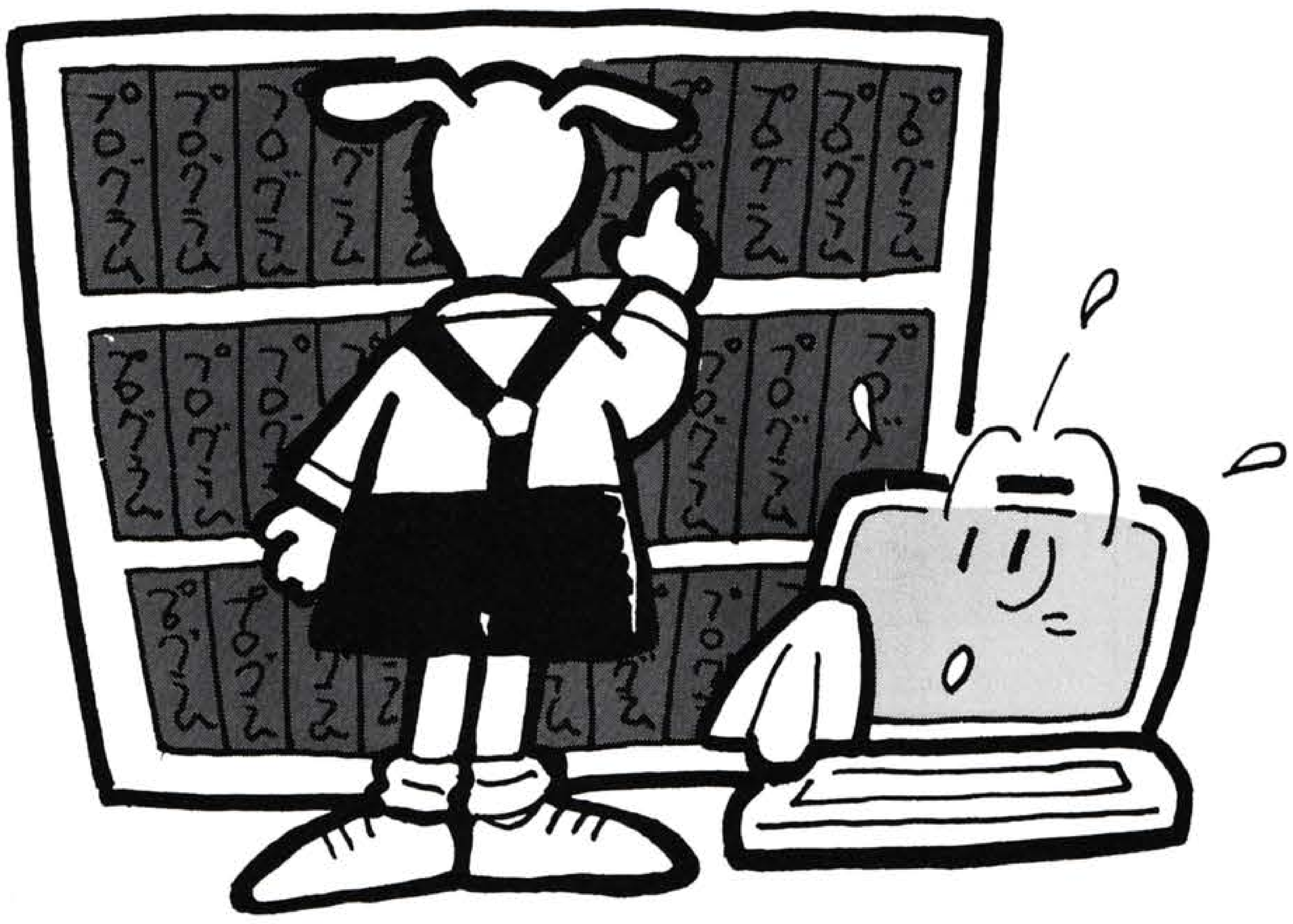


一度入力されたプログラムは、右のどれかの操作をするまで、パソコンが覚えています。

1．電源スイッチを切る
2．リセットボタンを押す
3．忘れるように命令する

ですから、プログラムを実行させる命令を入力するだけで、同じプログラムを何度でも繰り返し実行させることができます。

入力したプログラムは、フロッピーディスクに**「保存」**することができます。
保存したプログラムを次に使うときは、あらためてキーボードから入力する必要がありません。
保存したプログラムをパソコンの内部に**「呼び出し」**することにより、再び実行させることができるようになります。
（プログラムの保存と呼び出しについては、 34 ページから説明しています。）



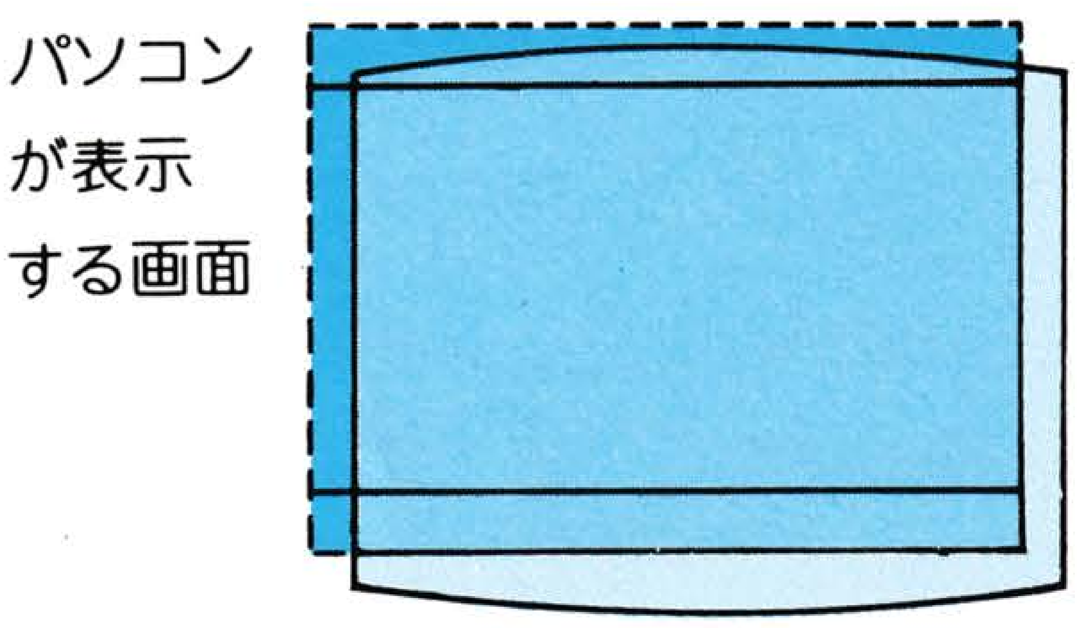
パソコンに直接命令してみよう(ダイレクトモード)

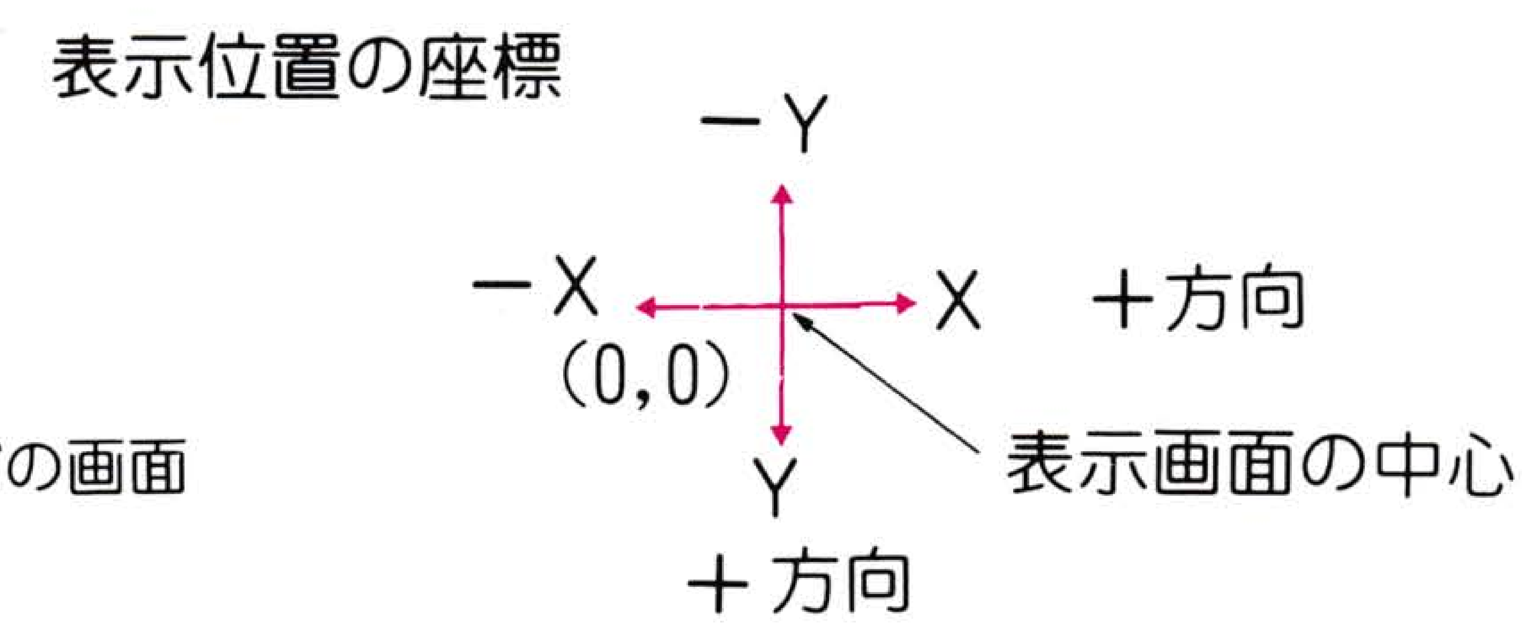
プログラムモードでパソコンを使う場合でも、
プログラムの入力や修正、実行や保存などの操作はダイレクトモードで行います。
ダイレクトモードでのパソコンの操作をもう少し練習しましょう。

画面の位置を合わす(SET ADJUST)

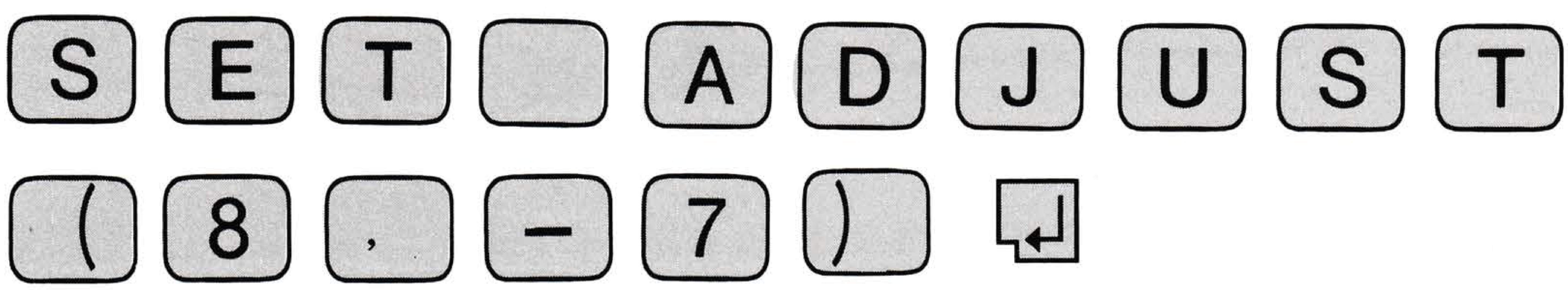
「**SET　ADJUST**」は、画面の表示位置を変更する命令です。パソコンの表示する画面が、ブラウン管の真ん中ではなく一方にずれて表示されるときに使用します。

BASICが初めてスタートしたときには、画面は（0，0）の位置に表示されます。





次のように入力して見ましょう。
（□は[SPACE]を、[－]はマイナスをあらわします。）



SET　ADJUST（X，Y）

（－7～＋8の数字）
※小数点以下は無視されます。

画面が右上に動いたのがわかりましたか？
SET　ADJUST命令は、左のような基本の形で入力します。
・**Xの部分には、**
画面の左右方向の位置をあらわす数字を入れます。
・**Yの部分には、**
画面を上下方向の位置をあらわす数字を入れます。

画面を初めの位置に戻すには、SET　ADJUST（0，0）↵と入力してください。

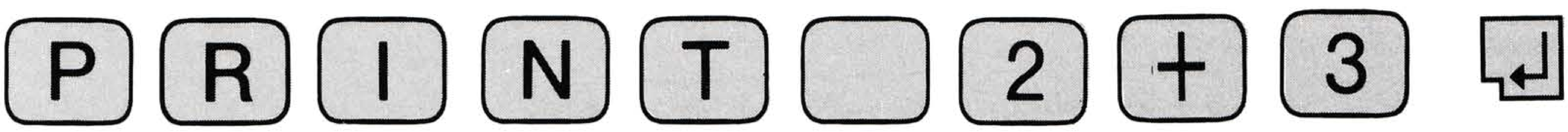
Xの部分を現在より大きい数字にすると右に、小さい数字にすると左に画面が移動します。

Yの部分を現在より大きい数字にすると下に、小さい数字にすると上に画面が移動します。

X、Yの部分の数字を変えて、パソコンの表示する画面がテレビのブラウン管の真ん中にくるように調整しておきましょう。これで、次にBASICをスタートしたときにも、設定した位置に画面が表示されます。（ただし、パソコンの乾電池が消耗したときや乾電池を入れ替えたときは、もう一度設定し直してください。）

画面に計算の答えや文字を表示する(PRINT)

PRINT（プリント）は、画面に文字や数字を表示させる命令です。
次のように入力してみましょう。




```
PRINT  2+3
 5
Ok
■
```

画面には左のように表示されますね。
PRINT命令の後ろに計算の式を書くと、計算の答えを画面に表示します。

BASICで計算に使う記号（**演算子**）の中には、学校などで習う記号とは形の違うものがあります。よく使われる演算子の書きかたを下の表にまとめておきます。

学校で教わる演算子	＋	−	×	÷	( )	X^2	**例**　((2+3)×5-6^2)÷3
BASICの演算子	＋	−	＊	／	( )	X^2	**例**　((2+3)*5-6^2)/3

上の表以外の演算子は、154ページに載っています。

画面に文字を表示するときは、表示したい文字を **「"」（ダブルクォート）** で囲みます。
このような、ひと続きの文字を **「文字列」** と呼びます。
次のように入力してください。

P R I N T [ ] " M S X " ↵

```
PRINT "MSX"
MSX
OK
■
```

画面には、次のように表示されます。
「"」（ダブルクォート）で囲んだ文字列が、そのまま表示されましたね。

メモ：「"」で囲んだ文字は、たとえ数字や式であっても、文字列としてあつかわれます。

《**ちょっとひと言**》

命令のつづりをまちがえて入力したときなどに、下のような文字が画面に表示されて、まちがえたことを教えてくれます。

```
Syntax error
```

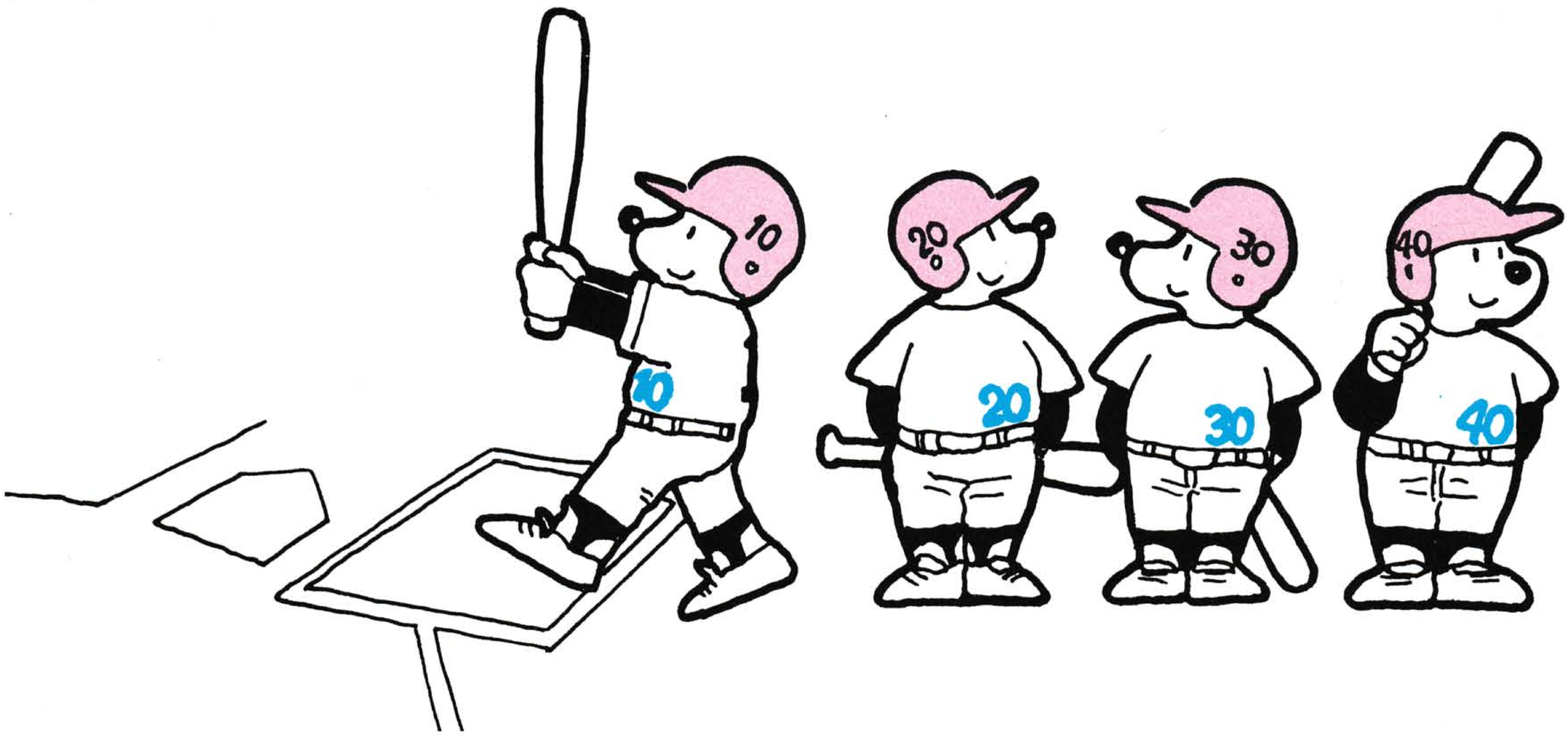
このような表示を**エラーメッセージ**と呼びます。

正しく命令を入力したつもりでいるときにいきなりエラーメッセージが表示されると、誰でもビックリして不愉快になります。
しかし、ＢＡＳＩＣでプログラムの作りかたを覚えようとする人にとって、エラーメッセージは大切な味方です。
エラーメッセージにはいくつもの種類があり、起きたエラー（まちがい）の内容に応じたメッセージ（知らせ）が表示されます。つまり、表示されたエラーメッセージの意味を調べることによって、何がいけなかったのかをある程度知ることができるのです。

表示されたエラーメッセージの意味は、207ページの「ＢＡＳＩＣエラーメッセージ表」を参照すれば分かります。
エラーメッセージが表示されてもイライラしたりせず、うまく利用してＢＡＳＩＣを覚えるために役立ててください。

プログラムに挑戦しよう

プログラムは、いくつかの命令をパソコンに分かる形式でまとめたものです。
BASICの命令のひとつひとつは小さな働きしかもっていません。しかし、たくさんの命令を組み合わせ、命令を実行する順番を指定し、パソコンが連続して働けるようにしてから「始め！」と号令をかけてやることによって、パソコンに大きな仕事をさせることができます。
つまり、プログラムといってもむずかしいことではなく、ひとつひとつの命令を実行させる順番に並べたものにすぎません。
命令を実行させる順番は、それぞれの命令を書いた行の先頭に付けた番号（行番号）で指定します。パソコンは、特別な命令を除いて、行番号が小さい行の命令から順に実行します。



それではここで、プログラムを作る作業の基本的なステップを説明しておきます。

① **パソコンにさせる仕事をハッキリ決める。**

② **その仕事をできるだけ細かな部分に分けて考える。**
次ページ下段のような図にして考えると便利です。

③ **それぞれの部分の仕事を、ＢＡＳＩＣの命令に置き換える。**
「資料編」の「ＢＡＳＩＣリファレンス表」を参照すると便利です。

④ **実行される順番に、命令に行番号を付けてプログラムにする。**
慣れないうちは、①〜④の作業はむずかしいものです。初めのうちはこの本や他のＢＡＳＩＣの解説書などに載っているプログラムをパソコンに入力し、実行させながら理解するようにしましょう。

⑤ **できたプログラムをパソコンに入力する。**
プログラムの入力のしかたについては、28ページから説明しています。

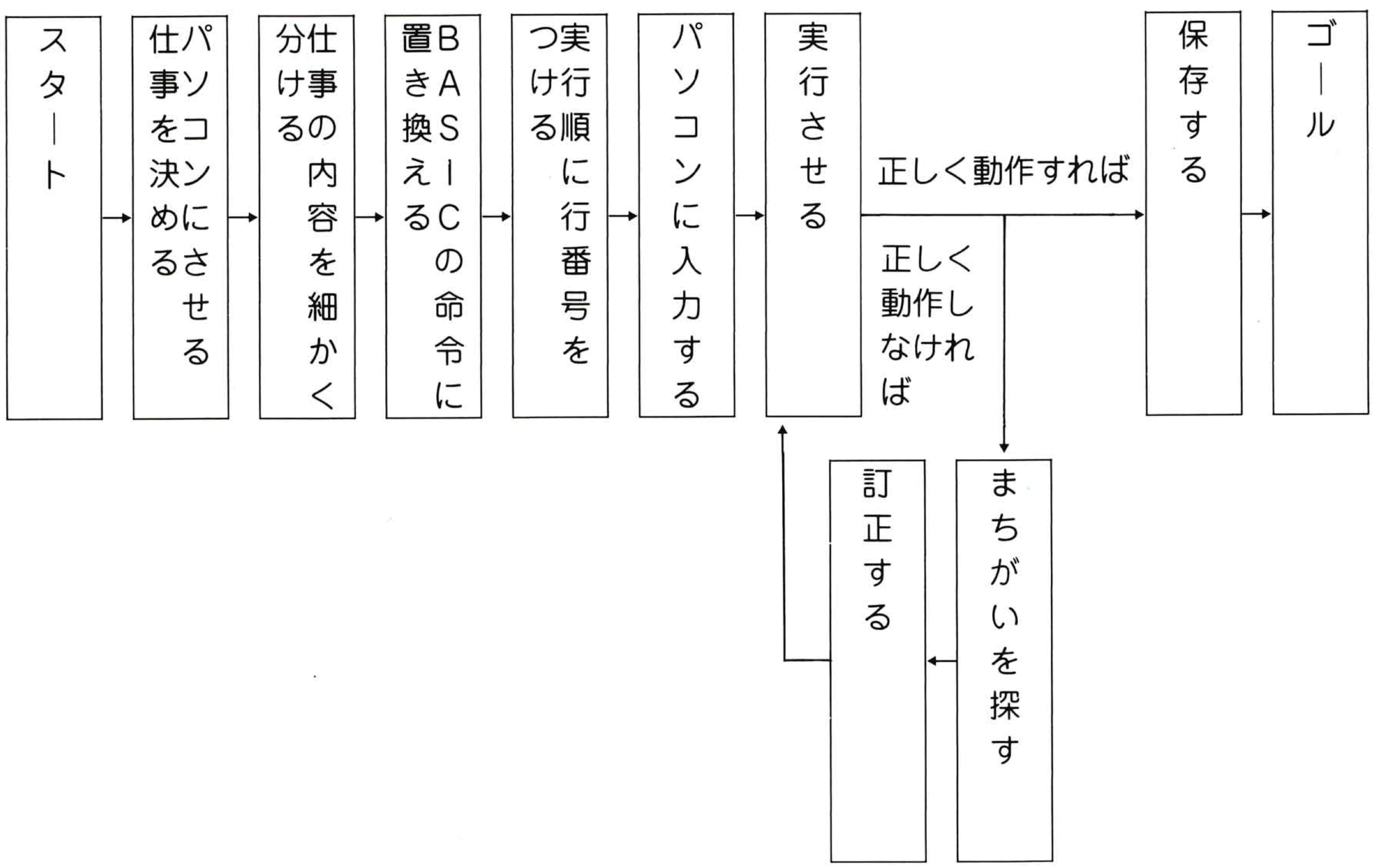
⑥ **プログラムを実行させる。エラーメッセージが表示されたり、正しく動作しなかったときには、プログラムをもう一度確かめ、まちがいを捜して修正する。**

プログラムの実行のさせかたと、エラーメッセージが表示されたときにどうしたら良いかは、30ページから説明しています。

⑦ **完成したプログラムをフロッピーディスクに保存する。**

プログラムの保存や呼び出しのしかたは、３４ページから説明しています。
プログラムの入力を途中で中断するようなときでも、それまで入力した部分を保存しておけば、あとで、保存した部分を呼び出して入力を続けることができます。

プログラムを作るときの仕事の流れ 例)



九九(くく)を表示するプログラム

それでは、かんたんなプログラムの例として、かけ算の九九を表示するプログラムを作ってみましょう。

パソコンにさせる仕事は、**「キーボードから入力された数を１〜９までの数にかけて、そのたびに式と答えを画面に表示すること」**と決めます。

パソコンがする仕事の内容を細かく分けると、次のようになります。

① **もとになる数の入力を求める。**
② **入力された数に１をかける。**
③ **かけ算の式と答えを画面に表示する。**
④ **②、③の仕事を、かける数を２〜９まで変えながら繰り返す。**

それぞれの仕事の内容をＢＡＳＩＣの命令に置き換えると、次のプログラム例のようになります。

行番号　命令　　　　　　　　　　　　　　**対応する仕事の番号**

```
10  INPUT "(1-9)";K          ①
20  FOR N=1 TO 9             ┐
30  L=K*N                    │ ②
40  PRINT K;"*";N;"=";L      │ ③
50  NEXT N                   ┘ ④
```

上のプログラム例の各命令は、次のような働きをします。

```
10  INPUT "(1-9)";K ——①の仕事に対応
                  変数
```

キーボードから入力された数値を、Ｋという名前の入れ物（**変数**）の中に入れます。
（変数については、４４ページでくわしく説明しています）
上のプログラム例では、"　"に囲まれた文字「（１−９）」と「？」を表示し、数値の入力を待ちます。

""の中の文字と、変数の名前は、自由に変えることができます。（上のプログラムでは、変数の名前は30行・40行と同じにすることが必要です。）""とその中の文字は省略することができます。

```
20 FOR N=1 TO 9 ────── ④の仕事に対応
       変数
50 NEXT N
        同じ変数
```

変数の数値を1，2，3，‥‥9と1つずつ増しながら、FOR命令とNEXT命令の間にある各命令を繰り返し実行します。Nの値が9を越えると、繰り返しをやめて次の行に進みます。

```
30 L=K*N ────── ②の仕事に対応
```

K×Nを計算し、計算結果を変数Lに入れます。

例) かけ算のもとになる数に5を入力したときは、FOR，NEXT命令の繰り返しの1回目ではK=5，N=1ですので5×1=5となり、変数Lには5が入ります。同様に2回目の繰り返しでは、5×2=10になり、Lには10が入ります。

```
40 PRINT K;"*";N;"=";L ── ③の仕事に対応
```

計算結果をKの値、＊、Nの値、＝、Lの値の順に画面に表示します。表示する項目の間に「;」がある場合は、前の項目に続けて次の項目を表示します。表示が終わると、次に文字が表示される位置はすぐ下の行の左端になります。

例)　K=5，N=1，L=5の場合

```
5 * 1 = 5
```

メモ： ●行番号は1、2、3…でもかまいませんが、あとで行を追加したりするときのために、10、20、30…と10間隔でつけるのがふつうです。

●FORとNEXTについては、48ページでくわしく説明しています。

それではこのプログラムを入力し、実行してみましょう。入力のしかた、実行のしかたは次のページから説明しています。

プログラムを入力しよう

新しくプログラムを入力する前に

プログラムを入力するときには、まず、ダイレクトモードで次の命令を入力して実行させます。

[N] [E] [W] [↵]

NEW（ニュー）命令は、現在パソコンが覚えているプログラムをすべて忘れさせる命令です。
パソコンがあるプログラムを覚えている状態のままで別のプログラムを入力してしまうと、2つのプログラムがひとつに混ざり合ってしまいます。
いまの場合はパソコンが覚えているプログラムは無いはずですが、プログラムの入力の前には必ずNEW命令を実行させる習慣を付けてください。

プログラムの入力のしかた

- **必ず行番号を行の先頭につけ、それに続けて命令などを入力します。**
- キーを押しまちがえたときの修正のしかたは、下図を参照してください。
- 1行が長いときには、入力を続けているあいだにカーソルが画面の右端から下の行の左端に移動することがありますが、かまわずに入力を続けます。
- **1行分の入力が終わるたびに、忘れずに [↵] を押します。**（ただし、画面で2行以上になるときは、途中の行で押してはいけません。）このとき、カーソルは必ずその行のどこかにあることが必要です。

修正のしかた		操作の手順
文字の上書き（まちがった文字を入力したとき）		CAS□ ⇨ CAS ⇨ [L] ⇨ CLS
文字の削除（不要な文字を入力したとき）	[BS]	CALS□ ⇨ CALS ⇨ [BS] ⇨ CLS（カーソルの前の文字を消す）
	[DEL]	CALS□ ⇨ CALS ⇨ [DEL] ⇨ CLS（カーソルのある位置の文字を消す）
文字の挿入 [INS]（文字を入力し忘れたとき）		CS□ ⇨ CS ⇨ [INS] ⇨ CS ⇨ [L] ⇨ CLS ⇨ [INS] ⇨ CLS

たとえば、このプログラムの10行を入力するときには、次のようにキーを押すことになります。（下の □ は、［SPACE］をあらわします。）

［1］［0］［ ］［I］［N］［P］［U］［T］［ ］［"］
［(］［1］［−］［9］［)］［"］［;］［K］［↵］

26ページのプログラムを、上のやりかたで入力してください。

行番号を付けて命令を入力すると、↵ を押しても命令は実行されません。
つまり、行番号を付けた命令を入力されると、パソコンは自動的にプログラムだと判断するわけです。

まちがいやすい文字に気をつけよう!

プログラムを入力するとき、よく起こすのはキーの打ちまちがいとリストの読みまちがいです。
次の文字は、特にまちがいやすいので注意してください。

・「**O（オー）**」と「**0（ゼロ）**」
・「**I（アイ）**」と「**l（小文字のエル）**」と「**1（いち）**」
・「**.（ピリオド）**」と「**,（カンマ）**」
・「**:（コロン）**」と「**;（セミコロン）**」
・「**S（エス）**」と「**5（ご）**」
・「**−（ハイフン、マイナス）**」と「**ー（カタカナの長音）**」

他には、文字の抜け（特に「"」「（　）」など）や、つづりのまちがいに気をつけてください。

プログラムの実行—失敗しても当たり前

それでは、「かけ算の九九を表示するプログラム」を実行させてみましょう。

プログラムを実行する(RUN)

RUN（ラン）は、プログラムを実行させる命令です。　（ F5 を押しても同じです。）

プログラムの入力が終わったら、次のように入力してください。

R U N ↵

画面に「（1－9）？■」と表示されたら、1～9までのどれかの数字キーを押してから ↵ キーを押します。

```
RUN
(1-9)? 3
 3 * 1 = 3
 3 * 2 = 6
 3 * 3 = 9
 3 * 4 = 12
 3 * 5 = 15
 3 * 6 = 18
 3 * 7 = 21
 3 * 8 = 24
 3 * 9 = 27
Ok
■
```

例として、3を入力してみます。
プログラムが正しく動いていれば、画面に左のように表示されます。

もし、画面に何も表示されなかったり、何かエラーメッセージが表示されたら、プログラムを入力したときにどこかをまちがえています。
でも、ガッカリする必要はありません。次のページからプログラムの修正のしかたを説明しますので、あきらめずに正しく動くまで訂正してください。

プログラムを訂正する

入力したプログラムを画面に表示させてまちがいを探し、訂正します。

画面や紙の上に表示されたプログラムを**「リスト」**と呼びます。
リストの全体を表示させる場合は、次のように入力します。

[L] [I] [S] [T] [↵]

ＬＩＳＴ（リスト）は、入力したプログラムのリストを画面に表示させる命令です。
リストが長すぎてひとつの画面に表示し切れないときは、最初の行から順に画面の外に消えていきます。これを**「スクロールする」**と呼びます。

・**スクロールを一時停めたいときは、**[STOP]を押します。
このときは、リストを見てまちがいを探すだけで、プログラムの訂正はできません。再び[STOP]を押すと、スクロールを再開します。

・**スクロールを中止させたいときは、**[CTRL]を押しながら[STOP]を押します。

その他、
・**特定の１行だけを表示させたいとき**　……　ＬＩＳＴ　２０[↵]
上の例では、２０行だけを表示します。

・**ある範囲の行を表示させたいとき**　……　ＬＩＳＴ　２０－４０[↵]
２０行～４０行のすべての行を表示します。

・**先頭から途中の行までを表示させたいとき**　ＬＩＳＴ　－３０[↵]
先頭の行から３０行までを表示します。

・**途中から最後の行までを表示させたいとき**　ＬＩＳＴ　３０－[↵]
３０行からあとのすべての行を表示します。

[－]（マイナス）と[ー]（カタカナの長音）をまちがえないよう注意してください。ＢＡＳＩＣの命令では、**マイナス**だけを使用します。

エラーメッセージが表示されたときは

まちがいの内容によっては、プログラムの実行が中断され、画面にエラーメッセージが表示されてダイレクトモードに戻ることがあります。
この状態を**「エラーになる」**といいます。
プログラムを実行中にエラーになったときは、エラーメッセージの後ろにエラーになった行の行番号が表示されます。

例) Syntax error in 30

この場合、表示された番号の行にまちがいがあるときと、他の行のまちがいが原因になってその行でエラーになったときの二通りがあるので注意してください。

表示されるエラーメッセージは、エラーの種類を大まかにしめしています。それぞれのエラーメッセージの意味は、207ページの「ＢＡＳＩＣエラーメッセージ表」にのっています。まちがいを探す参考にしてください。

まちがいを見つけたら

まちがいが見つかったら、画面のリストを正しく書き換え、↵ を押します。

- ↵ は、１行書き換えるたびに押してください。
- ↵ を押すときには、カーソルを書き換えた行に合わせておいてください。他の行にカーソルを動かしてから ↵ を押しても、プログラムは修正されません。
- 書き換えているあいだにカーソルが次の行まで届いてしまったときは、↵ を押す前に、DEL を使って次の行を画面から完全に消してしまってください。カーソルが次の行に届いたままで ↵ を押すと、２つの行がくっついてしまいます。

訂正が終わったら、ＬＩＳＴ命令を使って確かにプログラムが訂正されたかどうか確かめます。書き換えたとおりになっていたら、ＲＵＮ命令を入力してもう一度プログラムを実行させてみます。まだ正しく動かないようなら、他にもまちがいがありますから、もう一度まちがいを探して訂正してください。

《ちょっと一言》

大きなプログラムの修正に便利な命令

ＤＥＬＥＴＥ　（デリート）……指定された行を消去します。

例）

ＤＥＬＥＴＥ　５０　（ 50行を消去）

ＤＥＬＥＴＥ　１００－２００　（100行～200行のすべての行を消去）

ＲＥＮＵＭ　（リナンバー）……行番号を付け替えます。

例）

ＲＥＮＵＭ　（10行から10間隔で行番号を付け直す）

ＲＥＮＵＭ　１００，３０　（30行からあとのすべての行を、100行から10間隔の行番号に付け直す）

プログラムの保存と呼び出し

パソコンは、電源スイッチが切れた瞬間に、入力されたプログラムを忘れてしまいます。そこで、入力したプログラムを保存しておく方法を説明します。
一度プログラムを保存しておけば、いつ電源スイッチを切ってもだいじょうぶ。
いつでも、保存しておいたプログラムをパソコンの中に呼び出すことができるからです。

ディスクへの保存と呼び出しに必要なもの

・フロッピーディスクドライブ（ただし、ドライブ内蔵のパソコンでは必要ありません）
・フロッピーディスク（3.5インチ２ＤＤまたは3.5インチ１ＤＤと呼ばれるもの）
（※接続するフロッピーディスクドライブの機種によっては、接続用のカートリッジ（フロッピーディスクコントローラ）を別に買い求めることが必要です。）

どこで手に入れる?

フロッピーディスクドライブやディスクは、パソコンを専門に売っているお店（パソコン専門店）で売られています。また一般の電器店でも、パソコンを売っているお店なら手に入ることがあります。ただし、これらの機器を買い求めるときには、次のことをお店の人に確かめてください。

フロッピーディスクドライブについては、
・**「ＭＳＸ用」か？**
　ＭＳＸ用以外のフロッピーディスクドライブは接続できません。
・フロッピーディスク**コントローラが内蔵**されているか？
　もし内蔵されていない機種なら、コントローラを別に買い求めることが必要です。

ディスクについては、
・**「3.5インチ２ＤＤ」**（両面）または**「3.5インチ１ＤＤ」**（片面）と呼ばれる規格のものか？
　それ以外の規格のディスクを使うと、プログラムの保存や呼び出しができません。

実際の操作は?

以降のページでは、次のことを説明します。

●新しいディスクを使う前に必要な準備 ……………36ページから

●ディスクへのプログラムの保存のしかた …………38ページから

●ディスクからのプログラムの呼び出しかた ………39ページから

次のことは、機種による違いがあるので、この本では説明しません。
これらについては、あらかじめ、パソコンとフロッピーディスクドライブの説明書を読んでおいてください。

・フロッピーディスクドライブとパソコンの接続のしかた

・フロッピーディスクドライブを接続したときの電源スイッチの入れかた

・ディスクの出し入れのしかた

新しいディスクを使う前に必要な準備

お店で買って来たばかりの何も保存されていない新しいディスクを、この本では、
「生(なま)ディスク」と呼ぶことにします。
(パソコンの付属ソフトのフロッピーディスクや、ゲームやワープロなどのソフトが入ったディスクは、生ディスクではありません。)

生ディスクは、そのままでは使えません。まだ、**「フォーマット」**(または初期化)と呼ばれる準備をしていないからです。

フロッピーディスクの中には、**「磁気シート」**と呼ばれるごく薄い円盤状のフィルムが入っています。保存したプログラムやデータは、すべてこのシートに磁気の強弱の形で書き込まれます。
生ディスクの磁気シートは白紙のような状態で、何も目印になるものがありません。そこで生ディスクを使う前には、ちょうどノートのケイ線や原稿用紙のマス目のように、プログラムなどを書き込む位置を決める目印を書き込むことが必要です。これが「フォーマット」です。

> **ご注意:** すでに何かが保存されているフロッピーディスクをフォーマットすると、プログラムやデータが消えてしまいますから注意してください。

フォーマットのしかたは、次ページから説明しています。

ディスクをフォーマットする(CALL FORMAT)

以下の説明は、次のどちらかの場合について説明しています。
・内蔵ドライブ付きのパソコンでドライブを増設していない場合
・内蔵ドライブ無しのパソコンでドライブを1台だけ接続している場合
また、すでにBASICがスタートしているものとします。

1

CALL　FORMAT [↵] と入力します。

CALL　FORMAT（コール・フォーマット）は、ディスクをフォーマットする命令です。

2

画面に下のように表示され、ドライブの指定を求めてきます。

```
Drive name? (A, B) ■
```

画面の表示は、パソコンと、接続しているフロッピーディスクドライブの機種およびドライブの合計数によって違います。

3

[A] を入力します。(ドライブAを指定)

画面に次のように表示され、フォーマットの種類を聞いてきます。

```
1 - 1 side, double track
2 - 2 sides, double track

? ■
```

画面の表示は、パソコンと接続しているフロッピーディスクドライブの機種によって違います。

4

[2] を押します。(3.5インチ2DDディスクのとき。1DDなら [1] を押す)

画面に次のように表示され、用意ができているかどうかの確認を求めてきます。

```
Strike a key when ready ■
```

上記の [1] からの操作を始める前に、次の操作をしてください。
フォーマットするディスクをドライブに差し込み、「FILES [↵] 」と入力してから、画面を見る。
・「Disk I/O error」と表示されたとき ………… [1] からの操作を始める。
・「File not found」と表示されたとき ………… フォーマット済みなので、ディスクを交換する。
・いくつかのファイルスペックが表示されたとき ….. フォーマット済みなので、ディスクを交換する。

5 ディスクのライトプロテクトタブが書き込み可能の位置にあることを確かめてから、フロッピーディスクドライブ（ドライブを増設しているときは、増設したドライブ）に、生ディスクを差し込み、⏎ を押します。
（文字キーや SPACE を押してもかまいません。）

6 フロッピーディスクドライブのアクセスランプが点滅して、フォーマットを始めます。
※フォーマットには、少し時間がかかります。必要な時間は機種によって違いますが、一般的には2～3分で終わります。

7 フォーマットが終わると画面に右のように表示され、アクセスランプが消えます。

```
Format complete
Ok
■
```

8 ディスク取り出しボタンを押し、フォーマットしたディスクを取り出します。

保存するプログラムの名前を決める(ファイルスペック)

プログラムを保存するときには、まずプログラムに名前を付けます。プログラムを呼び出すときには、その名前によって他のプログラムと区別します。この名前を**「ファイルスペック」**と呼びます。ファイルスペックは次ページのSAVE命令やLOAD命令などといっしょに入力します。

例）

"A:MYPROG1.BAS"

- A … デバイス名
- MYPROG1 … ファイル名
- . … ピリオド
- BAS … 拡張子

デバイス名の部分には、保存に使うフロッピーディスクドライブの**ドライブ名**と、**「：」（コロン）**を書きます。
・ドライブ名は、A～Gの英文字1文字で指定します。
・ドライブが1台だけの場合は、ドライブ名はAになります。
・使用するドライブがドライブAのときは、デバイス名を省略できます。
・ドライブが2台以上ある場合は、接続の方法と機種によって、ドライブ名が変わります。
一般的には、ドライブ1台内蔵のパソコンにドライブを1台増設したときには、増設したドライブがA、内蔵ドライブがCになります。

ファイル名は、8文字以内の英文字、数字、カタカナで自由に書きます。
・英字の大文字と小文字は区別されず、すべて大文字とみなされます。

拡張子の部分は、「．」（ピリオド）の後ろに3文字以内で自由に書きます。
（カタカナや記号、小文字については、ファイル名と同じです。）
・拡張子の使いみちは自由です。プログラムの分類や新旧を区別するのに便利です。
・拡張子は省略できます。ただし、拡張子を付けて保存したプログラムを呼び出すときは省略できません。

では、「かけ算の九九を表示するプログラム」を、前ページの例のファイルスペックでディスクに保存してみましょう。

プログラムをディスクに保存する(SAVE)

以下の説明では、すでにプログラムがパソコンに入力されているものとします。
プログラムを保存することを**「セーブ」**と呼びます。

1 フォーマット済みのディスクを、ライトプロテクトタブが書き込み可能の位置にあることを確かめてからドライブAに入れます。

2 次のように命令とファイルスペックを入力します。

SAVE "A:MYPROG1.BAS" ⏎

SAVE（セーブ）は、プログラムをディスクに保存する命令です。

3 ドライブAのアクセスランプが点灯し、セーブが始まります。
セーブが終わると、画面に「Ok」と表示され、アクセスランプが消えます。

4 アクセスランプが消えていることを確かめ、ディスク取り出しボタンを押して保存したディスクを取り出します。

プログラムをディスクから呼び出す(LOAD)

保存したプログラムをパソコンのメモリの中に呼び出すことを**「ロード」**と呼びます。
ＢＡＳＩＣのプログラムをディスクからロードするときには、次のようにします。
（以下の説明では、すでにプログラムをディスクに保存しているものとします。）

1

プログラムを保存したディスクをＡドライブに入れ、次のように入力します。

```
LOAD "A:MYPROG1.BAS" ↵
```

ＬＯＡＤ（ロード）は、ディスクに保存したプログラムをパソコンの中に呼び出す命令です。必ず保存したときと同じファイル名と拡張子を入力してください。デバイス名は、使用するドライブのドライブ名に変えることができます。

2

ドライブＡのアクセスランプが点灯し、ロードが始まります。
ロードが終わると「Ｏｋ」と表示され、アクセスランプが消えます。

3

ディスクを取り出すときは、アクセスランプが消えたことを確かめてからディスク取り出しボタンを押し、ディスクを取り出します。

これで、プログラムはパソコンの中に呼び出されています。
ＬＩＳＴ命令を使って確かめてください。

ファイルスペックを確かめる(FILES)

プログラムを保存してから時がたつと、保存したプログラムのファイルスペックを忘れてしまうことがあります。
このとき、まちがえて同じディスクに保存している他のプログラムのファイルスペックを使って保存したりすると、保存されていたプログラムがそのプログラムに書き換えられてしまいます。
プログラムを保存したり呼び出したりする前には、なるべくそのディスクに保存されているすべてのプログラムのファイルスペックを確かめるようにしましょう。

ＦＩＬＥＳ（ファイルズ）は、ディスクに保存されているすべてのプログラムやデータのファイルスペックを画面に表示させる命令です。

ＦＩＬＥＳ命令の基本の形は、次のようになります。

```
FILES "ドライブ名:"
```

ドライブ名を省略すると、ドライブAに入っているディスクを調べ、ファイルスペックの一覧を画面に表示します。リストの表示と同様、一度に画面に表示しきれないときは初めに表示されたファイルスペックから順に消えて行きます。
（STOPを押すと画面の表示を一時止めることができ再び押すと再開します。）

ディスクの入っていないドライブを指定すると、下のエラーメッセージが表示されます。

```
Disk offline
```

これで、ディスクへの保存と呼び出しに必要な命令の説明は終わりです。この他にも、ＢＡＳＩＣにはディスクに保存したプログラムを管理するための便利な命令がありますので、ついでにここで紹介だけしておきます。

・ＣＯＰＹ（コピー）　……… ディスクに保存したプログラムを、別のディスクにコピーします。（これを「バックアップをとる」と呼びます）

例）　COPY "A:PROG1.BAS" TO "B: "

※例のように入力すると、アクセスランプが点灯したあと、画面に下の表示が出ます。

```
Insert disk for drive B:      ←(またはA:)
and strike a key when ready █
```

この表示が出るたびにドライブの中のディスクを入れ替え、↵を押してください。コピーが終わると、「Ｏｋ」と表示されます。(入力したファイルスペックのファイルがコピーされます。)

・ＫＩＬＬ（キル）　………… ディスクに保存したプログラムを消去します。

例）　KILL "NOUSE.BAS"

・ＬＦＩＬＥＳ（エルファイルズ）ディスクに保存されているすべてのプログラムやデータのファイルスペックを、プリンタで印刷します。（プリンタが必要です）

例）　LFILES

・ＮＡＭＥ（ネーム）　……… ディスクに保存したプログラムの、ファイルスペックを付け替えます。

例）　NAME "PROG1.BAS" AS "PROG2.BAS"

（"PROG1.BAS"＝旧ファイル名　"PROG2.BAS"＝新ファイル名）

・ＲＵＮ　"　"（ラン）　…… ＲＵＮ命令に続けてフロッピーディスクに保存されているプログラムのファイルスペックを入力すると、そのプログラムを呼び出したあと、自動的に実行します。

例）　RUN "MYPROG1.BAS"

第2章

BASICは楽しみながら覚えるモノ。とにかく入力して動かしてみましょう。

体験！プログラミング

BASICの約束ごと …… 44
（変数、数値の限界、プログラムの1行、文と複文）
白いヨーヨー …… 46
（テキスト画面の座標、プログラムの変更）
数当てゲーム …… 50
（IF文、関係演算、論理演算、変数型）
文字も数値であらわせる！ …… 54
（文字コード、16進数）
謎の三角形 …… 56
（三角関数、READ，DATA）
記憶力ゲーム …… 58
（配列変数）



BASICの約束ごと

新しい命令を説明する前に、ぜひ知っておいて欲しいことをかんたんにまとめて説明します。
これらは、ＢＡＳＩＣでプログラムを作るときに必ず守らねばならない約束ごとです。

変　数

変数とは、プログラムの中などで使う数値や文字列を覚えておくための**「入れ物」**のことです。（26ページのＫ，Ｎ，Ｌがこれに当たります）

・変数名

変数には、２文字までの英文字または英文字と数字で名前を付けますが、次のようなルールにしたがってください。

① 先頭の一文字は英文字であること。
② 英文字の大文字と小文字は区別しない。
　「Ａ１」と「ａ１」は同じ変数とみなされる。
③ 区別するのは先頭から２文字だけで、３文字目からは無視される。
　「ＡＢ」、「ＡＢＣ」、「ＡＢ１」は同じ変数とみなされる。
　ただし、型宣言文字（⇨文字型変数）は別。
④ ＢＡＳＩＣの予約語（⇨ 194ページ）を含む名前は使用できない。

・文字型変数

文字列を覚えるための**文字型変数**は、数値を覚えるための**数値型変数**と区別するため、変数名の後ろに「$」（ダラー）を付けます。「$」のように、変数の型を区別するための文字を**「型宣言文字」**と呼びます。

例）　「Ａ$」、「Ｂ１$」、「ＡＢ$」

・変数に数値や文字列を覚えさせるには

数値型変数　　変数名＝数値または演算式

例）	Ａ１＝３５００	変数Ａ１に 3,500を入れる。
	ＰＴ＝Ａ１＋３００	変数ＰＴに（Ａ１の値）＋300の計算結果を入れる。
	Ｙ＝Ｙ＋2	変数Ｙに（現在のＹの値）＋２の計算結果を入れる。

文字型変数　　変数名＝文字列または文字列の式

例）	ＭＪ$＝″ＭＯＪＩ″	変数ＭＪ$に文字列ＭＯＪＩを入れる。
	Ｐ$＝Ｋ$＋″かい″	変数Ｐ$に（Ｋ$の値の文字列）と「かい」をつないだ文字列を入れる。

BASICで扱える数値の限界

有効桁数　：14桁まで。14桁を越える場合は、15桁目で四捨五入。

例）　123.456789012345 ——→ 123.45678901235

数値の範囲：10^{-64} ～10^{62}

極端に大きい数字や小さい数字は、指数表示になります。

例）

数式の表示	ＢＡＳＩＣの表示
1.23×10^{-30}	1.23Ｅ－30
456×10^{36}	4.56Ｅ＋38

画面の1行とプログラムの1行

プログラムでいう**「１行」**とは、ひとつの行番号に続いて入力されたすべての文字をさします。プログラムの１行には、行番号や空白などを含め、最大 254文字まで入力できます。したがってプログラムの１行は、画面上の１行におさまらず、２つ以上の行にまたがって表示される場合があります。
プログラムの１行の区切りが分かりにくくなったときは、ＬＩＳＴ命令でプログラムのリストを表示し、調べたいプログラム行にカーソルを合わせて ⏎ を押してください。このとき、カーソルが表示された位置が次のプログラム行の先頭です。

文と複文

１つの命令を実際に入力するときには、命令だけでなく、いっしょに変数名や演算子などを入力することがよくあります。プログラムでは、これらを含めて**「文」**と呼びます。（文は命令だけでもかまいません）プログラムの１行には、２つ以上の文を**「：」**（コロン）で区切って入力することができます。このような行を**「複文」**または**「マルチステートメント」**と呼びます。

例）

```
10 CLS
20 INPUT "にゅうりょく";A :PRINT A
30 PRINT "このフ゜ロク゛ラムは、いみか゛ないフ゜ロク゛ラムて゛す。"
```

文（CLS）

複文（20行）

プログラムの１行（30行）

白いヨーヨー

プログラムの入力や実行、それに保存と呼び出しのしかたが分かったら、いろいろなプログラムを入力して動かしたり、一部を変えてみるのが命令を覚える早道です。ここからは短いプログラムの例をあげ、出てくる新しい命令の使いかたをかんたんに説明します。



テキスト画面で白いヨーヨーを動かします。ヨーヨーが上下に動くたびに、音（ビープ音）が鳴ります。プログラムを止めるときは、CTRL を押しながら STOP を押します。
ヨーヨーとひもの表示には、グラフィック文字「●」と「|」を使います。グラフィック文字は GRAPH を押しながら文字キーを押して入力します。
表示するタテの位置は、繰り返しの回数を決める変数Yの値で指定しています。110行～150行の繰り返しでヨーヨーを下げ、160行～200行で上げています。

リストと解説

リスト	解説
`1 'save"ヨーヨー.bas"`	プログラムを保存するときのファイルスペック
`100 SCREEN 1:WIDTH 29:CLS`	画面の初期化と消去
`110 FOR Y=0 TO 20`	Yの値が20になるまで120～140行を繰り返す
`120   LOCATE 14,Y  :PRINT "|";`	ヨーヨーのひもを表示する
`130   LOCATE 14,Y+1:PRINT "●";`	ヨーヨーを表示する
`140 BEEP`	音を鳴らす
`150 NEXT Y`	Yの値を1増して110行に戻る
`160 FOR Y=20 TO 0 STEP -1`	Yの値が0になるまで170～190行を繰り返す
`170   LOCATE 14,Y  :PRINT "●";`	ヨーヨーを表示する
`180 LOCATE 14,Y+1:PRINT " ";`	前に表示したヨーヨーを消す
`190 BEEP`	音を鳴らす
`200 NEXT Y`	Yの値を1減らして160行に戻る
`210 GOTO 110`	110行にジャンプする

新しい命令の説明

REM（リマーク）

ＲＥＭまたは「′」（アポストロフィ）は、同じ行の後ろの文字を無視させる命令です。（１行）

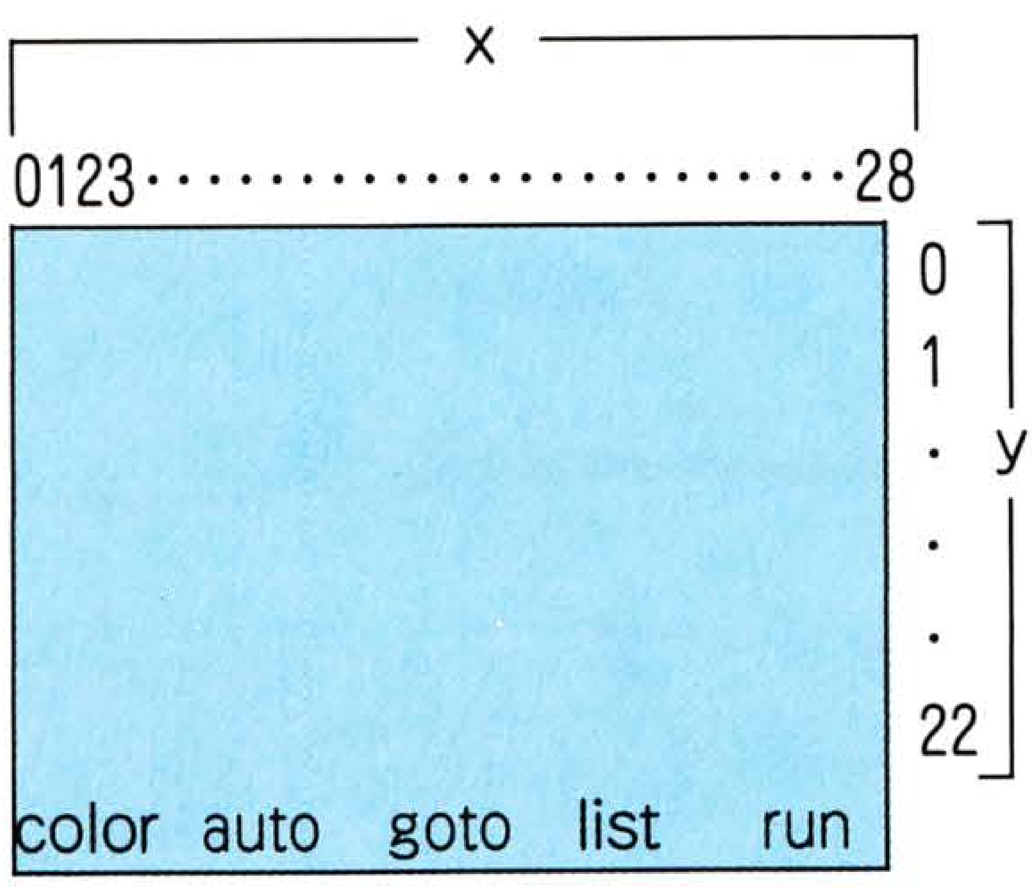
プログラムの中にちょっとした説明を入れるときなどに使います。必ず入力する必要はありません。
ただし、ＲＥＭ文を省略するときは、ＲＥＭや「′」だけでなくその行全体を省略します。
このプログラムの１行では、プログラムをセーブするときのファイルスペックを提案しています。

LOCATE（ロケート）

ＰＲＩＮＴ文などで文字を表示する位置を指定する命令です。

```
120 LOCATE 14, Y
```

14：ヨコ座標　Y：タテ座標



・ヨコ、タテの位置を１文字単位の数値で指定します。
・数値は０から始まり、画面の左上角が「０，０」にあたります。
・指定の順序は、常にヨコ座標を先に指定します。
・ヨコ座標、タテ座標とも、変数を使って指定できます。

《テクニック》

・ＳＣＲＥＥＮ文やＷＩＤＴＨ文を実行すると、画面の右端の値（28）が変わります。
・ＫＥＹ　ＯＦＦ命令を実行すると、画面の最下行も使用できます。（Ｙ：0〜23の範囲）

PRINT（プリント）

画面に文字や数値を表示する命令です。

```
130 PRINT "●";
```

"●"：文字または数値　;：区切り記号

・ＰＲＩＮＴ文の最後に**「;」**を付けて実行させると、次にＰＲＩＮＴ文を実行したとき、その後ろから続けて文字や数値が表示されます。
（ただし、ＬＯＣＡＴＥ文で表示位置を指定し直したときを除きます）
・ＰＲＩＮＴ文の最後に**「;」**や**「，」**を付けないで実行させると、次にＰＲＩＮＴ文を実行させたとき、その下の行の左端から表示されます。
・**「;」**や**「，」**を**「区切り記号」**と呼びます。区切り記号を使うことにより、一つのＰＲＩＮＴ命令で２つ以上の表示項目を表示することができます。
・表示項目の間を**「;」**で区切ると、数値や文字列を続けて表示します。（ただし、数値はそれぞれ前後にひと桁分の間隔が開けられます）
・表示項目の間を**「，」**で区切ると、数値や文字列の間を離して表示します。
（後ろの項目の表示位置は、ＳＣＲＥＥＮ文やＷＩＤＴＨ文によって変わります）

FOR, NEXT
（フォー、
ネクスト）

ＦＯＲ文とＮＥＸＴ文の間にある各文を、指定された回数だけ繰り返し実行する命令です。

```
110  FOR  Y=0  TO  20
          変数名 開始値    終了値
 :
150  NEXT  Y
           同じ変数名  ※変数名は数値型変数名に限る。
```

・ＦＯＲ文とＮＥＸＴ文は、必ず組み合わせて使います。対応するＦＯＲ文を実行しないうちにＮＥＸＴ文を実行させるとエラーになります。
・ＦＯＲ文とＮＥＸＴ文の変数名は、同じであることが必要です。
・ＦＯＲ文を実行すると、変数の値を開始値にして次の文を実行します。
・ＮＥＸＴ文を実行するたびに、変数の値に１を加えてから終了値と比較し、開始値～終了値の範囲を越えていなければＦＯＲ文の次の文から繰り返します。
・変数の値が開始値～終了値の範囲を越えると、繰り返しを終えてＮＥＸＴ文の次の文に進みます。

```
160 FOR Y=20 TO 0 STEP -1
         開始値  終了値   増分値
```

・ＦＯＲ文でＳＴＥＰ（ステップ）と増分値を入力したときは、ＮＥＸＴ文を実行したときに１を加える代わりに増分値を加えます。
・増分値は、負（マイナス）の値でもかまいません。このときは、開始値を終了値より大きくします。
・開始値、終了値、増分値は、変数でも指定できます。
・ＮＥＸＴ文の変数名は省略できます。省略したときは、最後に実行したまだ終了していない繰り返しのＦＯＲ文の、次の文から繰り返します。

GOTO
（ゴートゥー）

行番号で指定された行にジャンプし、その行の先頭の文から順に実行する命令です。

```
210  GOTO  110
           行番号
```

・行番号は、プログラム中に実在する行であることが必要です。省略はできません。
また、変数は使えません。

プログラムの変更

もっとたくさんのヨーヨーが表示されるよう、プログラムを変更しましょう。

プログラムを止めて、次のように入力してください。

```
115 FOR N=3 TO 23 STEP 5↵
135 NEXT N↵
165 FOR N=23 TO 3 STEP -5↵
185 NEXT N↵
```

現在プログラム中にない行番号で新たに入力した行は、プログラムに追加されます。

こんどは、次のように入力してください。

```
120 LOCATE N,Y:PRINT "|";↵
130 LOCATE N,Y+1:PRINT "●";↵
170 LOCATE N,Y:PRINT "●";↵
180 LOCATE N,Y+1:PRINT " ";↵
```

現在プログラム中にある行番号で入力すると、同じ行番号の現在の行が新しく入力した内容に書き換えられます。

入力が終わったら、ＬＩＳＴ命令でプログラムを確認してください。
正しく訂正できていたら、訂正前のプログラムとは異なる名前でセーブしておきましょう。

《ご注意》

・先にセーブしたプログラムと同じファイルスペックで同じディスクにプログラムをセーブすると、先にセーブしたプログラムが書き換えられて消えてしまいます。

数当てゲーム

パソコンが勝手に決めた数字を推理し、キーボードから答えを入力するゲームです。答えと正解が右下の図でタテまたはヨコの同じ列にあるときは、「おしい！」と表示されます。当たるまで答え続けて、何回で当てられるかを競ってください。



0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

途中で実行を中止したいときは、［CTRL］を押しながら［STOP］を押してください。

リストと解説

リスト	解説
`1 'save"カズアテ.bas"`	保存するときのファイルスペック
`100 SCREEN 1:WIDTH 29:CLS`	画面の初期化と消去
`110 R=INT(RND(-TIME)*25)`	乱数を使って正解を決める
`120 X=R MOD 5:Y=R ¥ 5:N=0`	正解がある位置の行と列を計算する
`130 '`	
`140 INPUT"よそう すうち (0-24) ";K:PRINT`	答えの入力を求め、変数Kに入れる
`150 IF K<0 OR K>24 THEN 130`	Kの値が範囲外なら、再び入力を求める
`160 N=N+1:X1=K MOD 5:Y1=K¥5`	答えの回数と答えの位置を計算する
`170 IF X=X1 AND Y=Y1 THEN200`	正解と答えの位置が同じなら200行へ
`180 IF X=X1 OR  Y=Y1 THEN A$="おしい!" ELSE A$="かんけいないもーん!"`	同じでないなら結果でA$の値を決める
`190 PRINT A$:PRINT:GOTO 130`	A$の値を画面に表示し､130行に戻る
`200 '`	
`210 PRINT "おめでとう";N;"かいで アタリ !!"`	答えが正しいときの表示
`220 END`	プログラムの実行を終了する

実行させると、推理した答えの入力を求めてきます。0～24の数字を入力してください。答えが合っていれば、何回目で当たったかを表示してプログラムの実行を終了します。

新しい命令の説明

関数

関数は、あたえられた材料（数値または文字列）を使って一定の仕事をし、その結果を返す小さな工場のような働きをするものです。

たとえば、ＩＮＴという関数は、「２．５」という数値を材料としてあたえられると、「２」という整数値を返します。

この材料としてあたえた数値や文字列のことを**「引数」**（ひきすう）と呼びます。

関数は、引数として数値をあつかう**「数値型関数」**と文字をあつかう**「文字型関数」**の２種類に大きく分かれます。

《ご注意》

関数は、命令とは異なり、それだけで文を作ることができません。

必ず命令か「＝」と組み合わせて文を構成します。

INT（インテジャー）（関数）

ＩＮＴは、あたえられた数値を整数にして返す関数です。

例）

```
    A＝ＩＮＴ（B）
数値型変数名      引数
```

・あたえられた引数の小数部分を切り捨て、整数部分を返します。

・引数は数値に限られます。定数（決まった数）でも変数でもかまいません。

RND（ランダム）（関数）

ＲＮＤは、数値をあたえられるたびに０以上１未満のでたらめな数値（乱数）を返す関数です。

例）

```
    A＝ＲＮＤ（B）
数値型変数名      引数
```

・引数が正（プラス）の数値のとき……プログラムをスタートさせてからＲＮＤ文を実行した回数によって返す乱数の値が決まる

・引数が０のとき……………………前回に返した乱数と同じ値を返す

・引数が負（マイナス）の値のとき……引数の値によって決まった乱数を返す

TIME
(タイム)
(システム関数)

システム変数とは、ＢＡＳＩＣが用意する特別な変数のことです。
ＴＩＭＥは、1/60秒ごとに１ずつ値が増すシステム変数です。

値の範囲は０～65535までで、65535の次は０に戻ります。プログラムやダイレクトモードで値を入れることもできますが、範囲を越えた値を入れるとエラーになります。

例）

```
TIME=0
Ok
PRINT TIME
 297
Ok
■
```

297：この値は、ＴＩＭＥ＝０を入力してからＰＲＩＮＴ　ＴＩＭＥを入力するまでの時間によって変わります。

このプログラムでは、ＲＮＤ関数の引数としてＴＩＭＥを負の値に変えたものが使われています。

MOD, ¥
(モジュロ、エン)
(演算子)

ＭＯＤは、割り算の「余り」を求める演算子です。
¥は、割り算の「商」を求める演算子です。

例）

```
A=5 MOD 2:B=5 ¥ 2
Ok
PRINT A;B
 1  2
Ok
■
```

・ＭＯＤ，¥で扱える数値は、－32768～＋32767までです。

このプログラムでは、乱数で決めた正解や入力された答えが図の中のどの行と列に当てはまるかの計算に使われています。

=
(イコール)
(関係演算子)

これまでの説明では、＝は変数にある値を入れるときに使いましたが、このプログラムの170行と180行では、＝を違う役目で使っています。これらの＝は**「関係演算子」**です。

関係演算子は、演算子の左右の値の関係を判定する記号です。
＝は左右の値が等しいかどうかを判定します。

AND, OR
(アンド、オア)
(論理演算子)

論理演算子は、数値またはその他のデータを一定の条件にしたがって操作する論理演算で、条件を設定する演算子です。

・ＡＮＤは、日本語の「かつ」にあたります。「Ａ　ＡＮＤ　Ｂ」は「ＡであるとＢ同時にＢである」という意味になります。

・ＯＲは、日本語の「または」にあたります。「Ａ　ＯＲ　Ｂ」は「ＡであるかまたはＢである」という意味になります。

このプログラムでは、２つの関係演算の結果が両方とも成立している場合の判定にＡＮＤを、どちらか片方だけが成立している場合の判定にＯＲを使っています。

IF~
THEN~
ELSE~
（イフ・ゼン
・エルス）

ＩＦ～ＴＨＥＮ～ＥＬＳＥ～（略してＩＦ文）は、ＩＦとＴＨＥＮのあいだに書かれた条件が成立するかどうかによって、プログラムの流れを変える命令です。

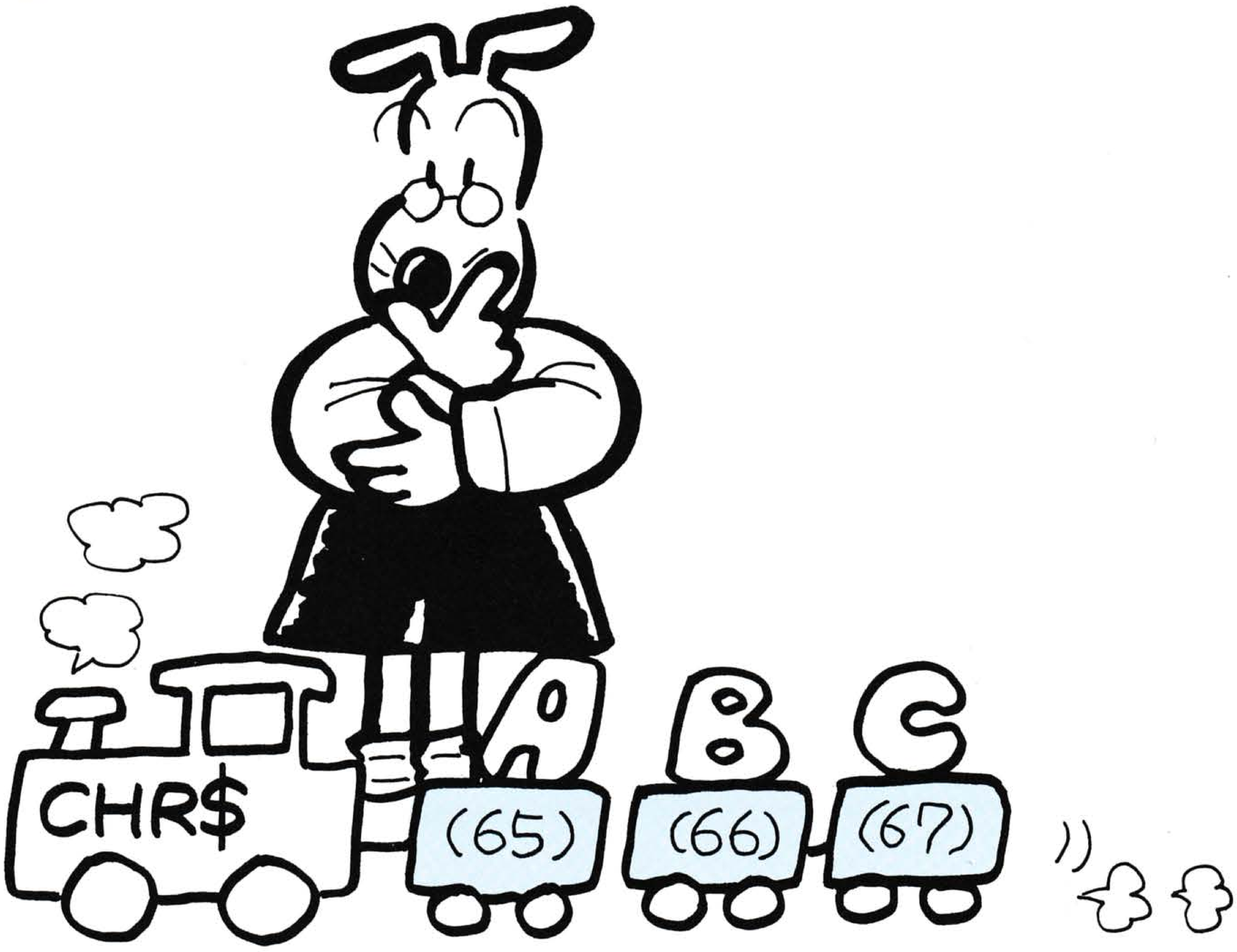
例）
```
IF A=B THEN 100 ELSE 200
```
条件（A=B）　命令または行番号（100）　命令または行番号（200）

・条件が成立すればＴＨＥＮの後ろの命令を実行するか、またはＴＨＥＮの後ろの行番号の行にジャンプします。
・条件が成立しなければＥＬＳＥの後ろの命令を実行するか、またはＥＬＳＥの後ろの行番号の行にジャンプします。
・ＥＬＳＥとその後ろの命令または行番号は、いっしょに省略できます。省略した場合に条件が成立しないときは、次の行に進みます。

170行のＩＦ文は、正解と入力された答えの両方の行と列が等しい場合は、200行にジャンプします。180行のＩＦ文は、行か列のどちらか片方だけが等しい場合とどちらも等しくない場合で、表示される文字列を変えています。

文字も数値で表せる?

ＢＡＳＩＣで使うすべての文字には、それぞれ異なる番号が割り当てられています。
この番号を「キャラクタコード」（または「文字コード」）と呼びます。
下のプログラムは、キーボードから入力できるすべての文字を、文字コードを使って画面に表示させるプログラムです。



リストと解説

プログラム	解説
`1 'save"ｾﾞﾝﾓｼﾞ.bas"`	ファイルスペック
`100 SCREEN 1:WIDTH 29:CLS`	画面の初期化と消去
`110 FOR A=33 TO 253`	「！」～「ん」の文字コードをＡの値に
`120   PRINT CHR$(A);`	Ａの値を関数の引数にして文字を表示
`130 NEXT A`	Ａの値を１増やして110行に戻る
`140 PRINT`	行を変える（改行）
`150 FOR A=65 TO 95`	グラフィックキャラクタのコードをＡの値に
`160   PRINT CHR$(1)+CHR$(A);`	グラフィックキャラクタの表示
`170 NEXT`	Ａの値を１増やして150行に戻る
`180 END`	プログラムの実行を終了する

※140行のように、ＰＲＩＮＴ命令の後ろに何も付けないで実行させると、改行だけが行なわれます。

新しい命令の説明

キャラクタコード表

ＢＡＳＩＣでは、すべての文字を番号であらわせます。この番号を「キャラクタコード」と呼びます。

キャラクタコードは、 174ページの**「キャラクタコード表」**に載っています。
この一覧表ではキャラクタコードを10進数ではなく、**16進数**であらわしています。16進数はＢＡＳＩＣでよく使いますから覚えておきましょう。
・10進数……１桁を０～９の10個の数字であらわす。何も記号を付けない。
・16進数……１桁を０～９と、10～15をあらわすＡ～Ｆの16個の数字であらわす。
※　10進数と区別するため、16進数の前には**「&Ｈ」**を付けます。

10進数の数字	0	1	2	3	…	9	10	11	12	13	14	15	16	17
16進数の数字	0	1	2	3	…	9	A	B	C	D	E	F	10	11

たとえば「Ｍ」のキャラクタコード「&Ｈ４Ｄ」は、10進数では16×４＋13で「77」になります。

CHR$（キャラクタ・ダラー）（関数）

数値であるキャラクタコードを対応する文字に変える関数です。

引数は当然キャラクタコードですが、10進数でも16進数でもかまいません。

例）
```
PRINT  CHR$(77);CHR$(&H4D)
MM
Ok
■
```

グラフィックキャラクタとヘッダ

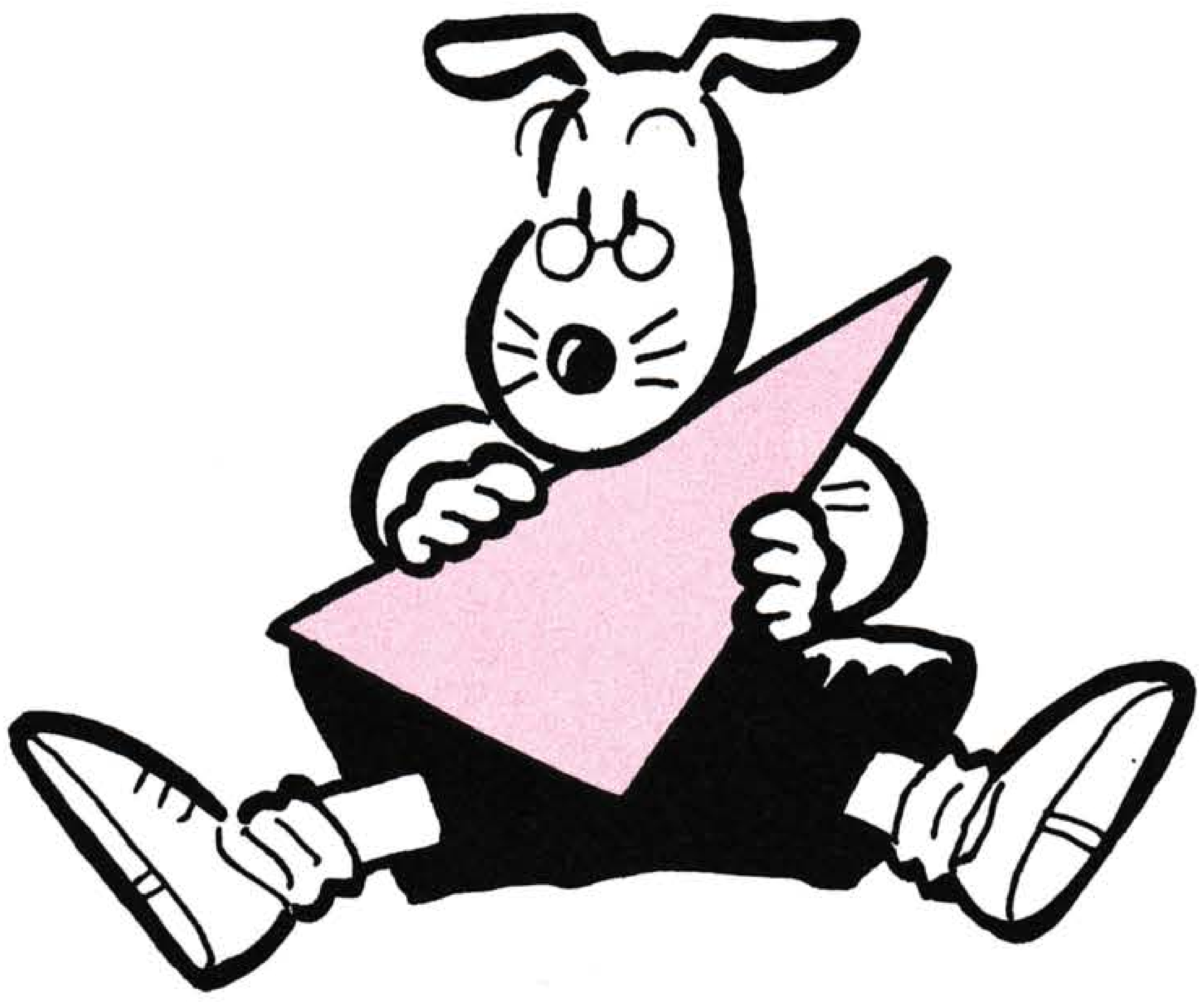
キーボードの(GRAPH)を押しながら入力する「月」などの文字を、「グラフィックキャラクタ」と呼びます。

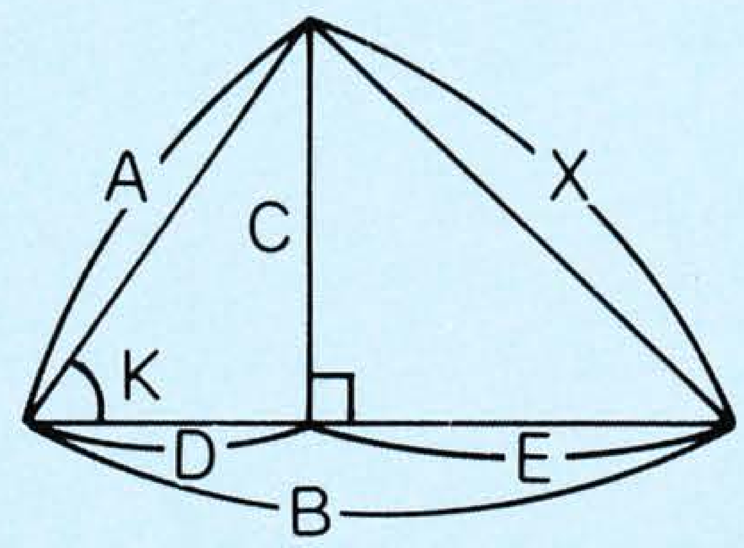
グラフィックキャラクタをキャラクタコードであらわすときは、キャラクタコードの前に**「グラフィックヘッダ」**と呼ばれる特別なコードを付けることが必要です。
グラフィックヘッダは、**「&Ｈ１」**に決められています。

例）
```
PRINT  CHR$(&H1)+CHR$(&H4D)
百
Ok
■
```

謎の三角形

二辺の長さとそのあいだの角度が分かっている三角形の、残りの一辺の長さを求めるプログラムです。





リストと解説

```
1 'save"ｻﾝｶｸ.bas"
100 SCREEN 0:WIDTH 40:CLS
110 P=3.14
120 READ A,B,K
130 F=K*P/180
140 C=A*SIN(F)
150 D=A*COS(F)
160 E=B-D
170 X=SQR(C^2+E^2)
180 PRINT "A=";A;"B=";B;"K=";
K
190 PRINT "X=";X
200 END
210 DATA 5,7,60
```

行	解説
100	画面の初期化（１行を40桁に）と消去
110	変数Ｐの値をπ（円周率）にする
120	ＤＡＴＡ文からデータを読み取り、各変数に
130	角度をラジアン単位に変換して変数Ｆに入れる
140	サイン関数で垂線Ｃの長さを求める
150	コサイン関数で辺Ｄの長さを求める
160	辺Ｅの長さを求める
170	目的の辺Ｘの長さを求める
180, 190	結果を表示する
200	プログラムの実行を終了する
210	辺Ａ，Ｂの長さと角度Ｋのデータ

実行すると、辺Ａの長さが５で辺Ｂの長さが７、その間の角度Ｋが60°の三角形の、残りの辺Ｘの長さを計算して三辺の長さを表示します。
210行の数値を書き換えて、いろいろな三角形の未知の辺の長さを求めることができます。

新しい命令の説明

READ DATA (リード、データ)

READは、プログラム中のDATA文から読み取ったデータを変数に入れる命令です。

必ずDATA文と組み合わせて使います。

```
120 READ A, B, K
         変数名 変数名 変数名
 ・
 ・
210 DATA 5, 7, 60
         データ データ データ
```

・データは、前から順に読み取られて変数の値になります。
・データの内容は、数値でも文字列でもかまいませんが、変数の型はデータと同じであることが必要です。
・READ文の変数の個数は、DATA文のデータの個数以下でないとエラーになります。
・READ, DATAは、ダイレクトモードでは使えません。

SIN COS (サイン、コサイン)(関数)

SIN, COSは、BASICに用意されている組み込み関数のひとつで、数学の授業で習う三角関数のサイン・コサインと同じ働きをします。

例)

```
A=SIN (B)
変数名   引数

A=COS (B)
変数名   引数
```

ただし、次の点に注意してください。
・引数は、ラジアンを単位にします。(πラジアンで180°になります)

SQR (スクェアルート)(関数)

数学の授業で習う「√￣」(ルート)にあたる関数で、引数の平方根を求めるのに使います。

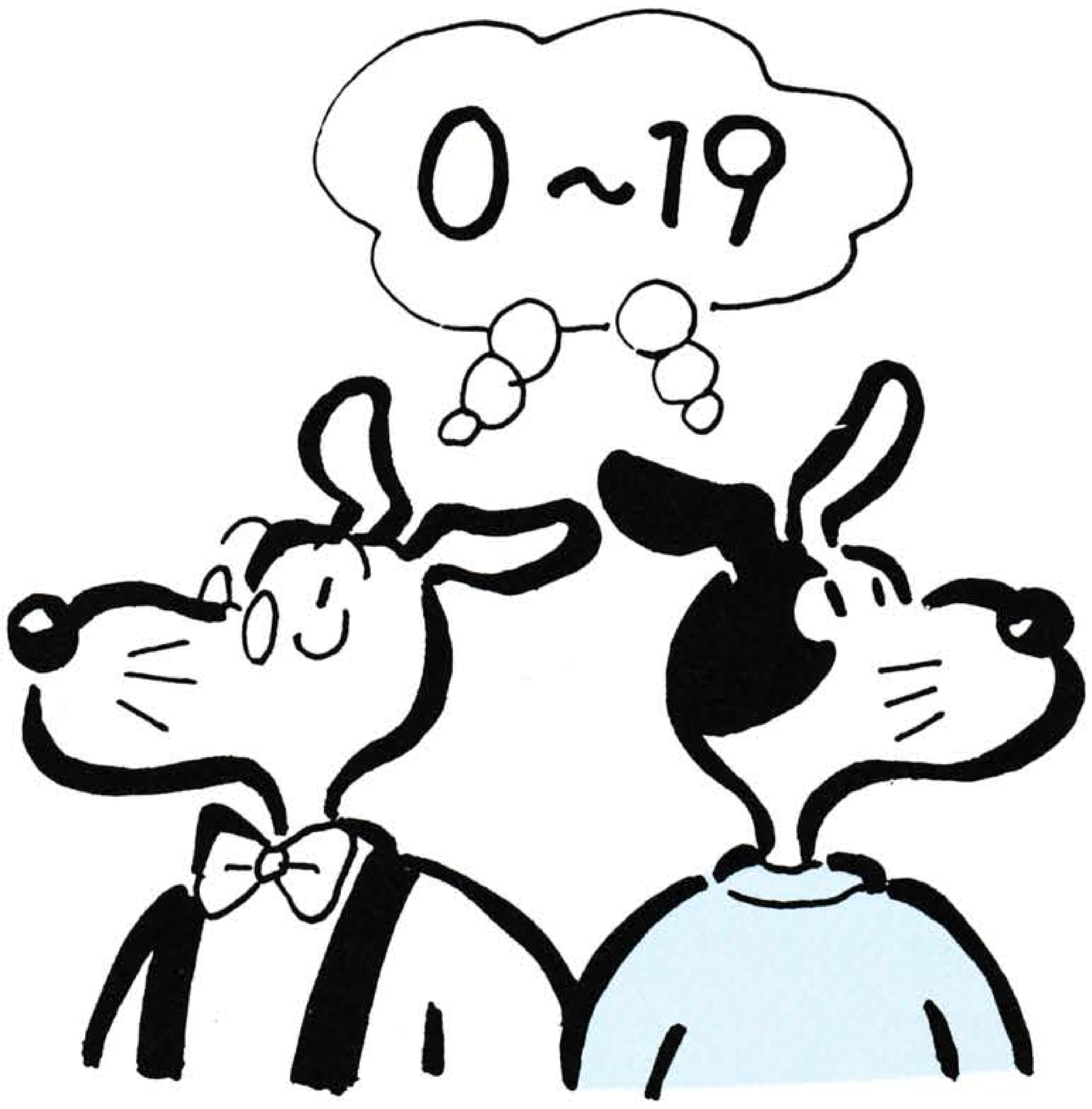
例)

```
A=SQR (B)
変数名   引数
```

記憶力ゲーム

二人で記憶力を競うゲームです。０～19までの数字を交互に１つずつ入力し、すでに入力されたのと同じ数字を入力してしまった人が負けです。自分と相手が入力した数字を覚えながら、できるだけデタラメな順番で数字を入力して相手を混乱させてください。

※メモを取ったりしたら反則ですよ！



リストと解説

```
1 'save"ｷｵｸﾘｮｸ.bas"
100 DIM M(19)
110 SCREEN 1:WIDTH 29:CLS
120 '
130 LOCATE 5,10:PRINT "
          "
140 LOCATE 5,10:INPUT "すうじ (0
-19)";N
150 IF N<0 OR N>19 THEN 120
160 IF M(N)=0 THEN M(N)=1:BEE
P:GOTO 120
170 '
180 FOR B=1 TO 10:BEEP:NEXT
190 LOCATE 7,12:PRINT"ざんねんで
した !"
200 END
```

行	解説
100	Ｍ（０），Ｍ（１），…Ｍ（19）の20個の入れ物（配列変数）を用意する
110	画面の初期化と消去
130	空白を上書きして前回入力した数字を消す
140	数字の入力を求め、変数Ｎに入れる
150	数字が範囲を越えていたらやり直し
160	まだ入力されたことがない数字なら、配列変数に１を入れて120行へ
170	すでに入力された数字なら次の行へ
180	ビープ音を10回鳴らす
190	負けたことを表示する
200	プログラムの実行を終了する

新しい命令の説明

配列変数

プログラムの中で同じ種類のたくさんのデータを扱うとき、いちいち別の変数に入れるのは大変です。そんなときに便利なのが配列変数です。

普通の変数では、ひとつの変数名にはひとつの数値や文字列（データ）しか入れることができません。配列変数は、各データに番号をつけることによって、ひとつの変数名にたくさんのデータを入れることができます。

データを操作するときには、配列変数名の後ろに操作したいデータの番号を付けて指定します。この番号を**「配列変数の添え字」**と呼びます。

例）　M（6）　B$（10）　C3（X）

添え字を変数にしてプログラムの中で値を変えることにより、操作するデータの指定を変えることができます。

DIM
（ディメンジョン）

配列変数を使用するときは、あらかじめ添え字の最大値を定義することが必要です。

そのための命令がＤＩＭです。

```
100 DIM M(19)
```

配列変数名　添え字の最大値

・ＤＩＭ文は、できるだけプログラムの先頭で実行させてください。
　プログラムの途中で実行させるとエラーになることがあります。
・１つのプログラム中で、同じ変数名でのＤＩＭ文の実行は、１回だけしかできません。

おまけのプログラム

ここまでガンバッテきた人にごほうびです。
いわゆる「ツルカメ算」の問題を解くプログラムです。

・使いかた

実行させると、まずツルとカメの頭の数の合計の入力を求め、次に足の数の合計の入力を求めます。

正しい数が入力されると、ツルとカメのそれぞれの数を表示します。

入力された数が正しくないときは、再び入力を求めます。

```
1 ' save"ツルカメ.bas"
100 SCREEN 1:COLOR15,4,7:CLS
110 LOCATE 5,7
120 INPUT "あたまのかず は";A
130 LOCATE 5,9
140 INPUT "あし のかず は";B
150 IF A<0 OR B<0 THEN 100
160 IF B MOD 2>0 THEN 100
170 IF A<B/4 THEN 100
180 K=B/2-A
190 T=A-K
200 LOCATE 5,12:PRINT"ツルは";T
210 LOCATE 5,14:PRINT"カメは";K
220 END
```

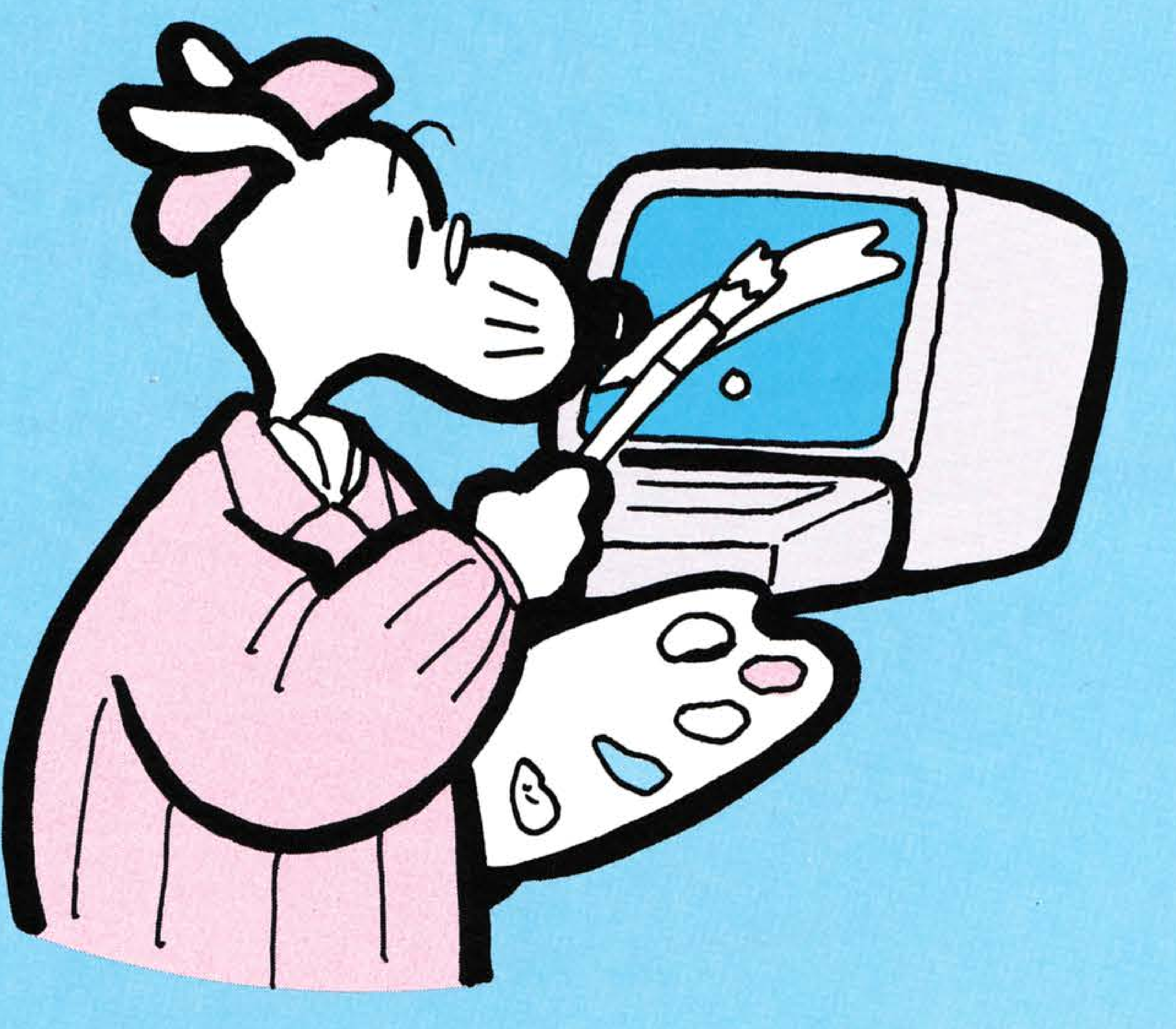
《ご注意》学校の宿題は自分で解きましょう。

第3章

ゲームにはグラフィックが欠かせません。プログラムで画面に絵を描きましょう。

お待たせ!グラフィック

グラフィック画面へ …………………………… 62
（スクリーンモード、SCREEN文）
直線、四角、円 ………………………………… 66
（グラフィック画面の座標、LINE, CIRCLE, PAINT）
いろいろな色 …………………………………… 68
（COLOR, COLOR=, 初期設定の色、パレット）
スクリーン8は宝石箱 ………………………… 70
（スクリーン8の色指定、PSET, PUT KANJI、長いプログラムの修正のしかた）
レモンを描こう…………………………………… 74
（楕円や扇形の描きかた）
スプライトって何だろう？ …………………… 76
（ゲームの主役、SPRITE$, PUT SPRITE、SCREEN, STICK）



グラフィック画面へ

お待たせしました。ここからは、画面に絵を描くグラフィック関係の命令について説明します。
まず最初は、これまで使ってきた文字表示用のテキスト画面と絵を表示するためのグラフィック画面の違う点について説明します。

グラフィック画面の種類

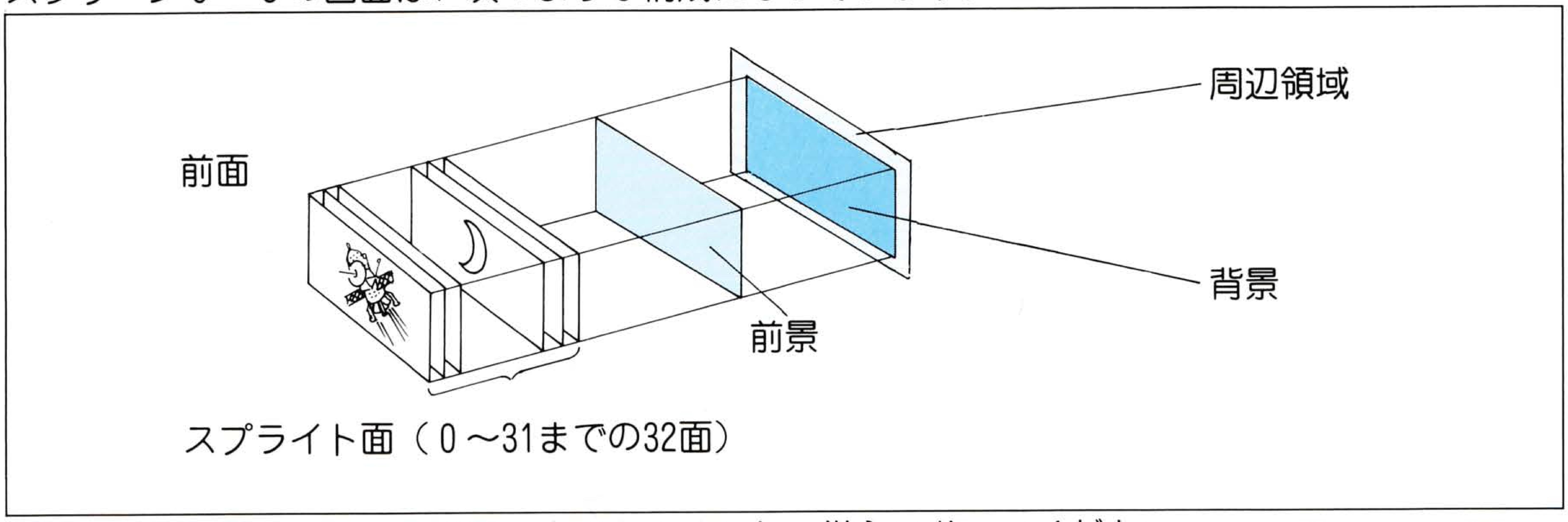
ＢＡＳＩＣの画面には、下の表のような種類があります。
（この表は、英数・かなモード時の各画面モードの仕様をしめしています）

モード	スクリーンNo.	文字表示	図形表示	表示色	色指定	スプライト
テキスト	0	初期値、1行39桁 最大80桁 ＊1	表示できない	512色中の16パレット中の2色	前景色 背景色 周辺色	表示できない
	1	初期値、1行29桁 最大32桁 ＊1	スプライトのみ可能（スプライト座標使用）＊2			8×8または16×16ドット 4個まで横方向の同じラインで同時表示可能
グラフィック	2	PRINT＃文で英数かなのみ表示 ＊3	横256×縦192ドット	512色中の16パレット	水平8ドットを1単位とする領域ごとに2色	
	3		横256×縦192ドット（64×48ブロック）＊4		1ブロックごとに色指定可能	
	4		横256×縦192ドット		水平8ドットを1単位とする領域ごとに2色	8×8または16×16ドット 8個まで横方向の同じラインで同時表示可能
	5	PUT KANJI文で漢字表示可 ＊3	横256×縦212ドット		1ドットごとに色指定可能	
	6		横512×縦212ドット	512色中の4パレット		
	7			512色中の16パレット		
	8		横256×縦212ドット	256色固定（スプライトは16色）		
自然画表示	10			自然画表示色、12,499色 ＊5		
	11	文字表示には向かない。		自然画表示色、19,268色 ＊5		
	12			自然画表示色、19,268色 ＊5		

＊1）英数・かなの表示数。
＊2）スプライト座標は、Xは－32～255、Yは－32～191（スクリーン5以上では、Yは－32～211）。
＊3）漢字モードでは、全画面モードで漢字表示可能。 漢字モードについては、98ページから説明。
＊4）1ブロックは4×4ドットの領域。
＊5）スクリーン11・12では、図形表示に256色分の色コードを指定できますが、色表示の方法が他の画面と異なるため、同じコードによる色指定をしても他の画面とは異なる色になったり、まったく表示されなかったりします。図形表示はスクリーン10までの画面モードでしてください。

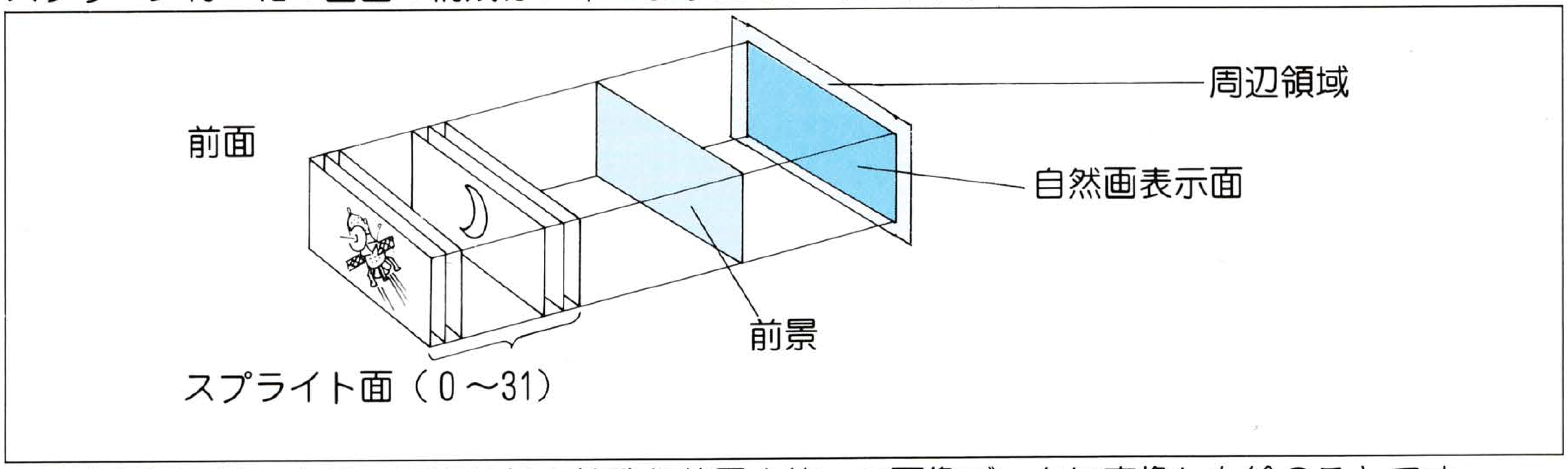
BASICの画面の構成

ＢＡＳＩＣの画面の種類を細かく分けると、スクリーン０～１２の１２種類になります。（スクリーン９はありません）これを、**「画面モード」**と呼びます。
スクリーン０～８の画面は、次のような構成になっています。



これらの名前はこれからの説明に出てきますから、覚えておいてください。
・ただしスクリーン０には、スプライト面はありません。

スクリーン10～12の画面の構成は、下のようになっています。



・**「自然画」**は、カラー写真などを特殊な装置を使って画像データに変換した絵のことです。スクリーン10～12の**「自然画表示面」**は、自然画を表示するための特別な方式による画像表示を行う面です。
・ＢＡＳＩＣでは自然画を作成することができません。
・スクリーン10で図形や文字を描くと、自然画表示面に表示されます。
・スクリーン11～12は自然画表示専用です。ＢＡＳＩＣで図形や文字を描くと、色が違ったり表示されなかったりします。

ＢＡＳＩＣがスタートしたときには、テキスト画面になっています。これをグラフィック画面に切り替えるには、次の命令を実行させます。

画面モードを切り替える(SCREEN)

ＳＣＲＥＥＮ（スクリーン）は、画面モードを切り替えたり、各種の設定を変更する命令です。

例）

ＳＣＲＥＥＮ　５

画面モード

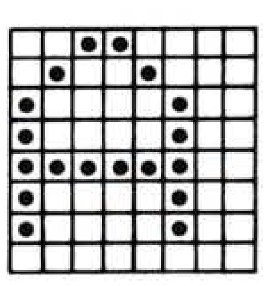
ＢＡＳＩＣがスタートしたときには、スクリーン１のテキスト画面が表示されます。
（ＳＥＴ　ＳＣＲＥＥＮ命令でスタート時の画面をスクリーン０に設定したときは除きます）

スクリーン０とスクリーン１の画面には、絵や図形を描くことができません。
これらの画面は**「テキスト画面」**と呼ばれ、文字の表示用です。
絵や図形を表示することができる画面を**「グラフィック画面」**と呼び、スクリーン２～１２がそれにあたります。そのうちスクリーン１０～１２は特に**「自然画表示モード」**と呼びます。

例）　ＳＣＲＥＥＮ　０　　（スクリーン０のテキスト画面に切り替える）
　　　ＳＣＲＥＥＮ　５　　（スクリーン５のグラフィック画面に切り替える）

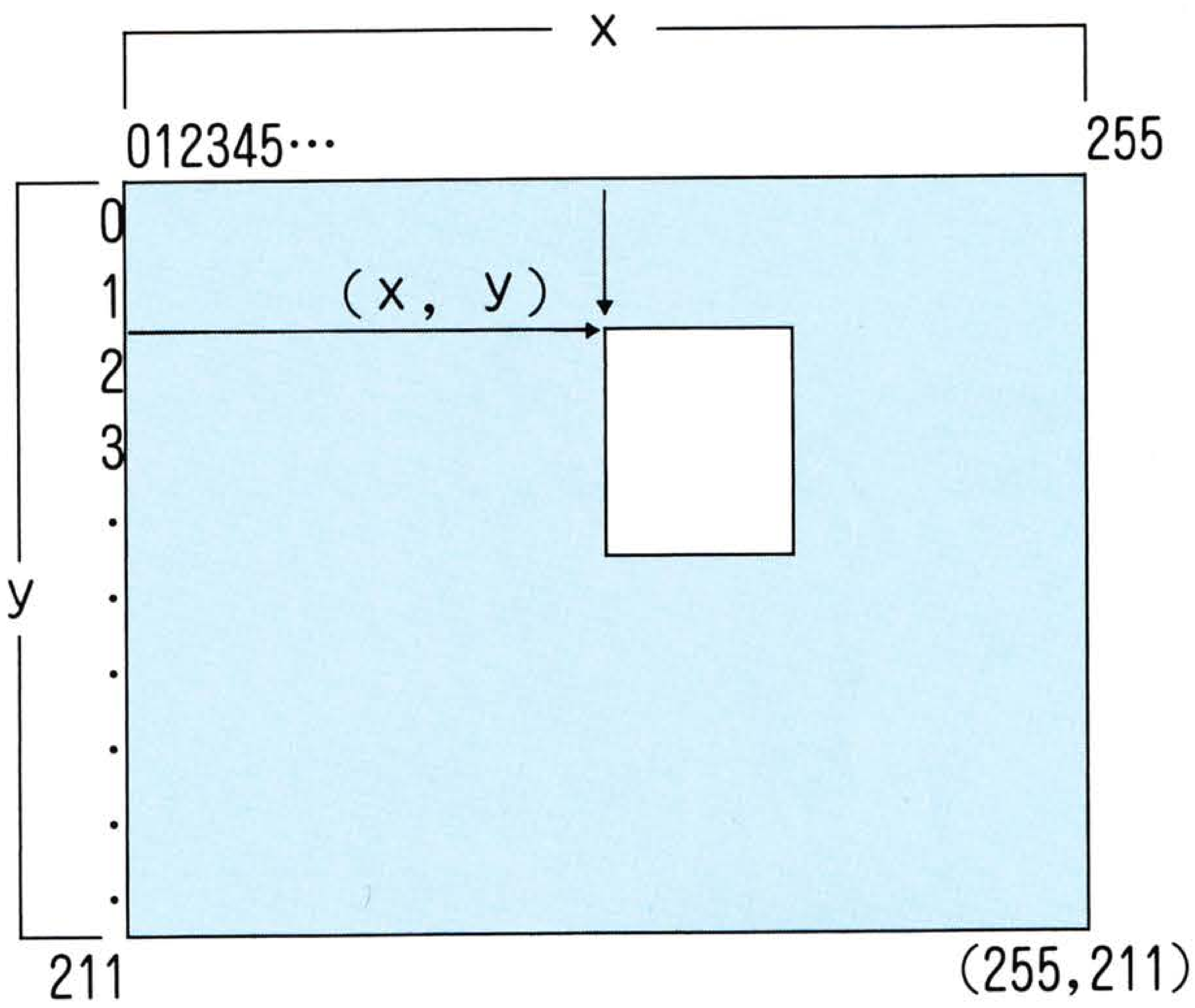
グラフィック画面での表示位置

テキスト画面、グラフィック画面の両方とも、画面に表示される文字や絵は、すべて「ドット」と呼ばれる小さな点を並べて表示されています。



テキスト画面では文字を表示する位置を指定するのに、「ヨコ何桁目、タテ何行目」というように１文字単位で指定しましたが、グラフィック画面では点を表示する位置を指定するのに「ヨコ何ドット目、タテ何ドット目」と、１ドット単位で指定します。各画面モードがタテ方向とヨコ方向にそれぞれ何ドットあるかは、62ページの表の「図形表示」に載っています。

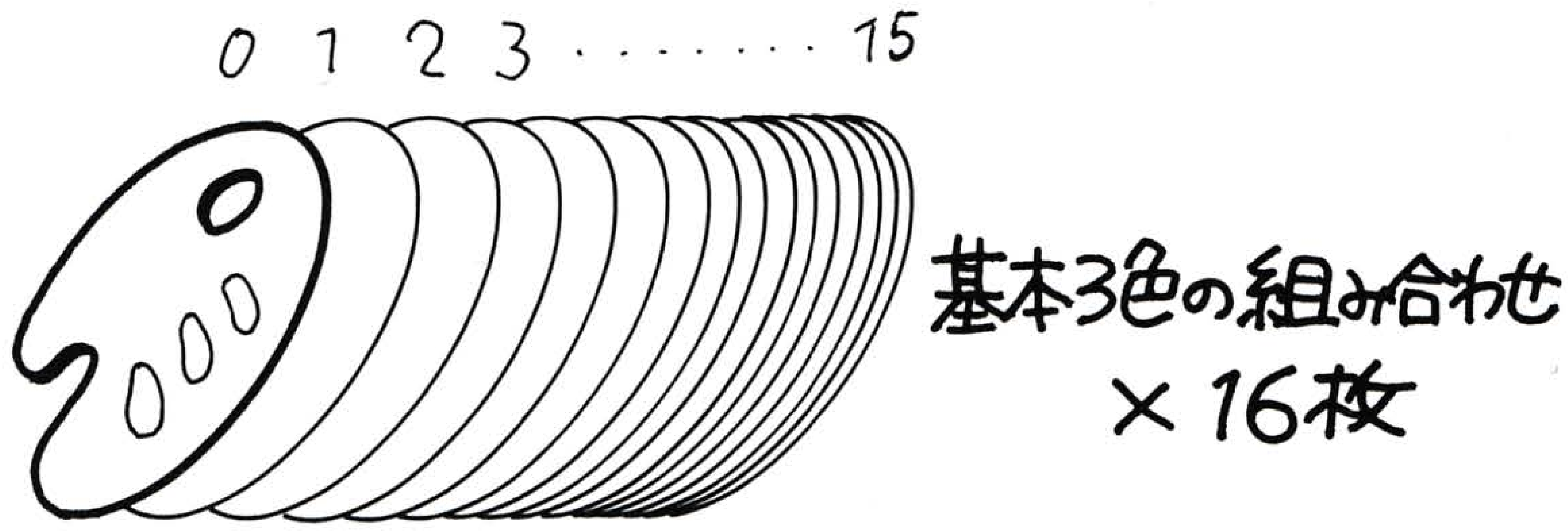
例） スクリーン５の位置座標



色の指定のしかた

スクリーン２～７で使用できる色の種類は、 512色あります。ただし、そのうち同時に表示できるのは、最大16色までです。（スクリーン６では４色まで。）

ちょうど油絵や水彩画を描く人が絵の具をパレットの上で混ぜ合わせて使いたい色を作り出すように、０～15までの番号に使いたい色を割り当て、実際に表示する時はこの番号で指定します。
この機能を**「パレット機能」**と呼び、番号を**「カラーコード」**と呼びます。つまり、スクリーン２～７までの画面には16枚のパレットがあるわけです。



ＢＡＳＩＣがスタートしたときには各カラーコードに下の色が設定されますので、これらの色だけを使うときは、あらためて各パレットに色を設定する必要はありません。

		赤　緑　青の各成分		
カラーコード	０：透明	（０，０，０）	８：赤	（７，１，１）
	１：黒	（０，０，０）	９：淡い赤	（７，３，３）
	２：緑	（１，６，１）	10：黄	（６，６，１）
	３：淡い緑	（３，７，３）	11：淡い黄	（６，６，４）
	４：濃い青	（１，１，７）	12：濃い緑	（１，４，１）
	５：青	（２，３，７）	13：紫	（６，２，５）
	６：濃い赤	（５，１，１）	14：灰	（５，５，５）
	７：水色	（２，６，７）	15：白	（７，７，７）

直線、四角、円

それでは、実際にグラフィック画面に図形を描いてみましょう。



リストと解説

リスト	解説
`1 'save"ﾁｮｸｾﾝ.BAS"`	
`100 SCREEN 5:COLOR:CLS`	画面モードをスクリーン5に切り替える
`110 LINE(100,10)-(150,60),10`	カラーコード10（黄）で直線をひく
`120 LINE( 30,10)-( 80,60),15,B`	カラーコード15（白）で四角を描く
`130 LINE(170,10)-(220,60), 3,BF`	カラーコード3（淡い緑）で四角を描き、中を塗りつぶす
`140 FOR C=1 TO 15`	コードを指定する変数の値を15回変える
`150   X=30:Y=70+C:D=4:D1=-D`	ジグザグを描き始める位置と変化値
`160   PRESET (X,Y)`	各線を最初に描き始める位置を指定する
`170   FOR M=0 TO 47`	直線を描くのを48回繰り返す
`180     X=X+D:D1=-D1:Y=Y+D1`	直線を描き終える位置の値を変える
`190     LINE -(X,Y),C`	前回の位置から今回の位置まで直線を描く
`200   NEXT M`	
`210 NEXT C`	
`220 FOR R=10 TO 55 STEP 5`	円の半径を5ずつ増しながら繰り返す
`230   CIRCLE (128,150),R,2`	カラーコード2（緑）で円を描く
`240 NEXT R`	
`250 PAINT (55,35),8,15`	カラーコード15（白）で囲まれた部分を8の色（赤）で塗りつぶす
`260 GOTO 260`	テキスト画面に戻らないようにする

新しい命令の説明

LINE（ライン）

ＬＩＮＥは、グラフィック画面に直線や四角を描く命令です。

直線を描くとき

```
110 LINE (100,10)-(150,60),10
```

始点の位置座標　終点の位置座標　カラーコード

四角を描くとき・・・・・・・・Ｂ

```
120 LINE ( 30,10)-( 80,60),15,B
```

対角をなす点の位置座標　四角形の指定

四角形を描き、同じ色で中を塗りつぶすとき・・・・・・・・ＢＦ

```
130 LINE (170,10)-(220,60), 3,BF
```

塗りつぶしの指定

PRESET（ポイント・リセット）

ＰＲＥＳＥＴは、描いた点を消すための命令です。

```
160 PRESET (X,Y)
```

ヨコ座標　タテ座標

このプログラムでは、次のＬＩＮＥ命令で線を描き始める位置（最終参照点）を設定するために使われています。（上の例の「Ｘ」,「Ｙ」は変数名です。）

CIRCLE（サークル）

ＣＩＲＣＬＥは、グラフィック画面に円を描く命令です。

```
230 CIRCLE (128,150),R,2
```

中心の位置座標　半径　カラーコード

・半径はドット単位で指定します。（上の例の「Ｒ」は変数名です。）

PAINT（ペイント）

ＰＡＩＮＴは、指定した位置を含む、境界色で囲まれた部分を領域色で塗りつぶす命令です。

```
250 PAINT (55,35),8,15
```

領域内の位置の座標　領域色　境界色

・指定した位置が境界色で完全に囲まれていないときは、画面全体が領域色で塗りつぶされます。

いろいろな色

ＢＡＳＩＣがスタートしたときに各カラーコードに割り当てられる色を画面で確かめるプログラムです。「－」（マイナス）の数値を入力すると、パレット機能の効果を確かめることができます。



実行させるとカラーコードの入力を求めます。０～15までの数値を入力すると、画面の周辺領域の色が変わります。（ただし、コード０は透明、コード１は黒）
マイナスの数値を入力すると、コード13の赤、緑、青各成分の入力を順に求めます。
０～７までの数値を入力してください。
表示される文字の色（コード13）が変化します。

プログラムを止めるときは、CTRL を押しながら STOP を押してください。

リストと解説

```
1 'save"イロイロなイロ.bas"
100 SCREEN 1:WIDTH 29:C1=7
110 COLOR:COLOR 15,4,C1:CLS
120 LOCATE 5,11:INPUT "COLOR
";C1
130 IF 15< C1 THEN 120
140 IF  0<=C1 THEN 110
150 CLS:COLOR 13,4,7
160 PRINT:PRINT "color 13 の R
 = ";:INPUT R
170 PRINT "color 13 の G = ";:
INPUT G
180 PRINT "color 13 の B = ";:
INPUT B
190 IF R<0 OR 7<R THEN 160
200 IF G<0 OR 7<G THEN 160
210 IF B<0 OR 7<B THEN 160
220 COLOR= (13,R,G,B)
230 GOTO 160
```

行	解説
100	画面を初期化し、変数Ｃ１に７を入れる
110	周辺色をＣ１の値にして画面を消去する
120	カラーコードの入力を求め、入力された値をＣ１に入れる
130	Ｃ１の値が15より大きければ、再入力を求める
140	Ｃ１の値が０以上なら、110行に戻る
150	Ｃ１の値が０以下なら、前景色をコード13に
160	カラーコードの13の赤成分の入力を求め、変数Ｒに入れる
170	カラーコードの13の緑成分の入力を求め、変数Ｇに入れる
180	カラーコードの13の青成分の入力を求め、変数Ｂに入れる
190–210	Ｒ，Ｇ，Ｂ各成分の値が０～７の範囲外なら、色を変更しないで再入力を求める
220	コード13の色を入力された成分比に変更する
230	160行に戻る

新しい命令の説明

COLOR（カラー）

次のように入力すると、画面の前景色、背景色、周辺色を変更します。

```
110  COLOR 15, 4, C1
           前景色 背景色 周辺色
```

・テキスト画面、グラフィック画面のどちらでも使えます。
・各色はカラーコードで指定します。使えるカラーコードの範囲は、画面モードによって変わります。
・次の場合には、文字などの表示が見えなくなります。
　→前景色と背景色に同じコードを指定したとき
　→前景色にカラーコード0（透明）を指定したとき
・ＢＡＳＩＣがスタートしたときには、「COLOR 15,4,7」に設定されています。
　F6（SHIFTを押しながらF1を押す）を入力するとこの設定になります。
・後述の漢字テキスト画面でＣＯＬＯＲ文を実行するとエラーになります。漢字テキスト画面では、ＣＡＬＬ　ＰＡＬＥＴＴＥ文を使用してください。（⇨178ページ）

次のように入力すると、指定されたカラーコードの赤、緑、青各成分を指定された比率にして、実際に表示される色を変えます。

```
220  COLOR=(13, R, G, B)
     カラーコード，赤の比率，緑の比率，青の比率
```

・この指定は、スクリーン0～7および10で有効です。スクリーン8にはパレット機能はありません。またスクリーン11～12では、指定した比率の色になりません。
・カラーコードは1～15までです。（スクリーン6では1～3）
・カラーコード0（透明）は、ＣＯＬＯＲ文では変更できません。
・赤，緑，青の各比率は、0～7までの数値で指定します。

スクリーン8は宝石箱

スクリーン8にはパレット機能はありませんが、そのかわり、256色まで同時に表示する機能があります。この機能を使って、たくさんの色で画面に絵を描いてみましょう。（このプログラムでは、あわせて、英数・かなモードのグラフィック画面に漢字を表示する機能も使っています）



プログラムを止めるときは、[CTRL] を押しながら [STOP] を押してください。

※漢字モードでの漢字の表示のしかたは、98ページから説明します。

リストと解説

```
1 'save"ﾎｳｾｷﾊﾞｺ.bas"
100 SCREEN 8:COLOR 255,0
110 LINE (50,100)-(150,160),2
55,BF
120 LINE (50,100)-(100, 50)
130 LINE          -(200, 50)
140 LINE          -(150,100),1
82
150 LINE          -(150,160),1
82
160 LINE          -(200,110),1
82
170 LINE          -(200, 50),1
82
180 PAINT (175,100),182
```

- 100：スクリーン8に：前景色を白に、背景色を透明に
- 110：カラーコード 255（白）で宝石箱の前面を描く
- 120～130：宝石箱の上面のへりを描く
- 140～170：宝石箱の側面をカラーコード182で描く
- 180：同じ色で側面を塗りつぶす

```
190 PUT KANJI (76,122),&H4A75
,28
200 PUT KANJI          ,&H4050
,28
210 PUT KANJI          ,&H4822
,28
220 A=RND(-TIME)
230 '
240 Y=INT(51+RND(1)*49)
250 X=151-Y+RND(1)*99
260 C=256*RND(1)
270 PSET (X,Y),C
280 GOTO 240
```

- 190〜210：漢字コード&H4A75,&H4050,&H4822の漢字（「宝」「石」「箱」）を赤で表示する
- 220：乱数系列を初期化する
- 240：Yに51〜99の範囲の乱数を入れる
- 250：Xに-46〜100の範囲の乱数を入れる
- 260：Cに0〜255の範囲の乱数を入れる
- 270：X，Yの位置にCの色で点を描く
- 280：240行に戻る

新しい命令の説明

スクリーン8

スクリーン8のカラーコードは、表示される色を次のような形式で直接にあらわしています。

緑色の成分……G（0〜7の8段階）
赤色の成分……R（0〜7の8段階）
青色の成分……B（0〜3の4段階）

計算方法

G×32＋R×4＋B＝カラーコード

したがって、スクリーン8で指定できるカラーコードは、0〜255の256種類になります。

・カラーコード0は、他の画面モードと同じ透明です。
・カラーコード255は、白です。

PUT
KANJI
(プット
・カンジ)

ＰＵＴ　ＫＡＮＪＩは、英数・かなモードで漢字をグラフィック画面に表示する命令です。

```
190 PUT KANJI (76, 122), &H4A75, 28
```

ヨコ座標　タテ座標　漢字コード　カラーコード

・グラフィック画面のスクリーン5～10でだけ使用します。

・ヨコ座標、タテ座標は、文字の左上の位置をあらわします。文字はタテ16ドット，ヨコ16ドットの範囲の中に表示されます。省略したときは、最後に実行した位置座標を使うグラフィック関係の命令で使用した位置（最終参照点）に表示されます。
ＰＵＴ　ＫＡＮＪＩを実行すると、最終参照点は16ドット右に移動します。

・**漢字コード**は取扱説明書の「ＪＩＳ漢字コード表」に載っています。（ただし、非漢字については、本書の219ページを参照のこと）漢字コード表は「キャラクタコード表」と同様に、ＪＩＳコードが16進数で表示されています。16進数は、前に「＆Ｈ」を付けて入力します。

・カラーコードは、使用する画面モードによって使えるコードの範囲が変わります。
カラーコードを省略すると、前景色で表示されます。

・表示される文字は、文字の形の描線とそれを囲む四角い部分で構成されています。

PSET
(ポイント
・セット)

ＰＳＥＴは、グラフィック画面に点を表示する命令です。

```
270 PSET (X, Y), C
```

ヨコ座標　タテ座標　カラーコード

・カラーコードは省略できます。省略すると、カラーコードは前景色になります。

・ＰＳＥＴを実行すると、最終参照点がその位置に移動します。

《テクニック》

70ページのサンプルプログラムでは、漢字が表示された部分の周囲が黒くなってしまいます。これはグラフィック画面に表示される文字が、文字の部分とその周囲の四角い部分でできているからです。文字の周囲の部分は、自動的に背景色（この場合は透明）になります。
文字の周囲を背景色にしたくないときは、次のようにプログラムを変更します。

```
190 PUT KANJI (76,122),&H4A75,28,TPSET
200 PUT KANJI          ,&H4050,28,TPSET
210 PUT KANJI          ,&H4822,28,TPSET
```

プログラムを実行すると、箱の絵の上に漢字の文字の部分だけが表示されます。

ＴＰＳＥＴは**「論理演算子」**と呼ばれるもののひとつで、ＰＵＴ　ＫＡＮＪＩ命令といっしょに使用すると、透明色（カラーコード0）の部分を表示させないようにします。

したがってＴＰＳＥＴを使用するときは、あらかじめ背景色をカラーコード0に設定してください。

レモンを描こう

画面に半分に切ったレモンを描くプログラムです。
だ円や扇形を描くＣＩＲＣＬＥ命令の便利な機能を使っています。

プログラムを止めるときは、CTRL を押しながら STOP を押してください。



リストと解説

```
1 'save"レモン.BAS"
100 P=3.14:X0=160:Y0=100
110 SCREEN5:COLOR 15,4,7:CLS
120 LINE (0,120)-(255,211),2,
BF
130 CIRCLE (X0,Y0),120,10,-.5
*P,-1.5*P,.75
140 PAINT (X0-1,Y0),10
150 CIRCLE (X0,Y0),90,15,,,2.
5
160 PAINT (X0,Y0),15
170 CIRCLE (X0,Y0),80,10,,,2.
5
180 PAINT (X0,Y0),10
190 FOR J=-2*P TO 0 STEP .8
200   CIRCLE(X0,Y0),80,15,J,J
+.4,2.5
210 NEXT J
220 GOTO 220
```

- 100：変数Ｐに円周率を、Ｘ0，Ｙ0に円の中心座標を
- 110：画面をスクリーン５にし、初期設定
- 120：緑色でテーブルの表面を描く
- 130：黄のだ円でレモンの胴体を描く
- 140：胴体を黄で塗る
- 150：白のだ円でレモンの断面を描く
- 160：断面を白く塗る
- 170：黄のだ円でレモンの中身を描く
- 180：中身を黄で塗る
- 190〜200：白でレモンの房をあらわす扇形を描く

新しい命令の説明

これまで説明を省略しましたが、ＣＩＲＣＬＥ命令には、だ円や扇形を描く機能もあります。

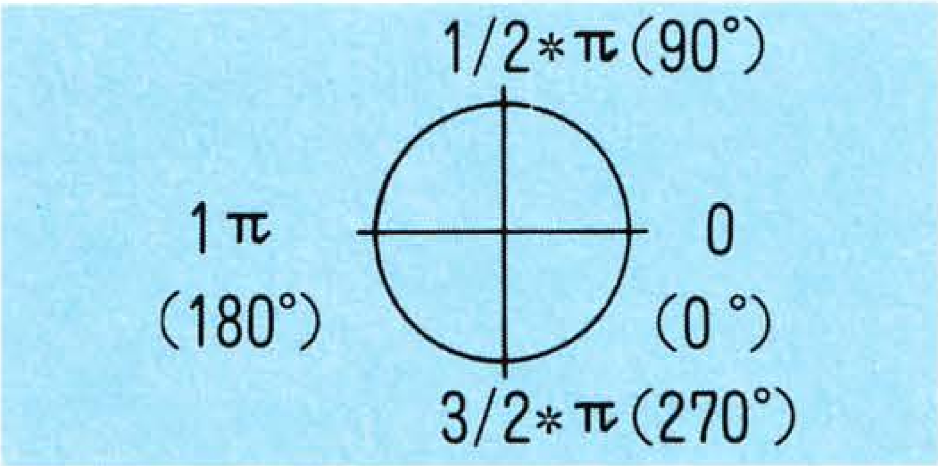
CIRCLE命令によるだ円や扇形の描きかた

```
130 CIRCLE  (X0, Y0), 120, 10, -.5*P, -1.5*P, .75
```

中心の位置座標　半径　表示色　開始角度　終了角度　比率

新しく出てきた項目について説明します。

・**開始角度，終了角度**は、円弧を描くときに入力します。角度の単位はラジアンです。180°が１π、360°が２πになります。（π＝3.14）



※常に円の右端が０になります。

※０から反時計回りに開始角度、終了角度の順に指定します。

開始角度，終了角度の数値の範囲は、－２π～＋２πまでです。

開始角度を省略したときは０、終了角度を省略したときは２πになります。

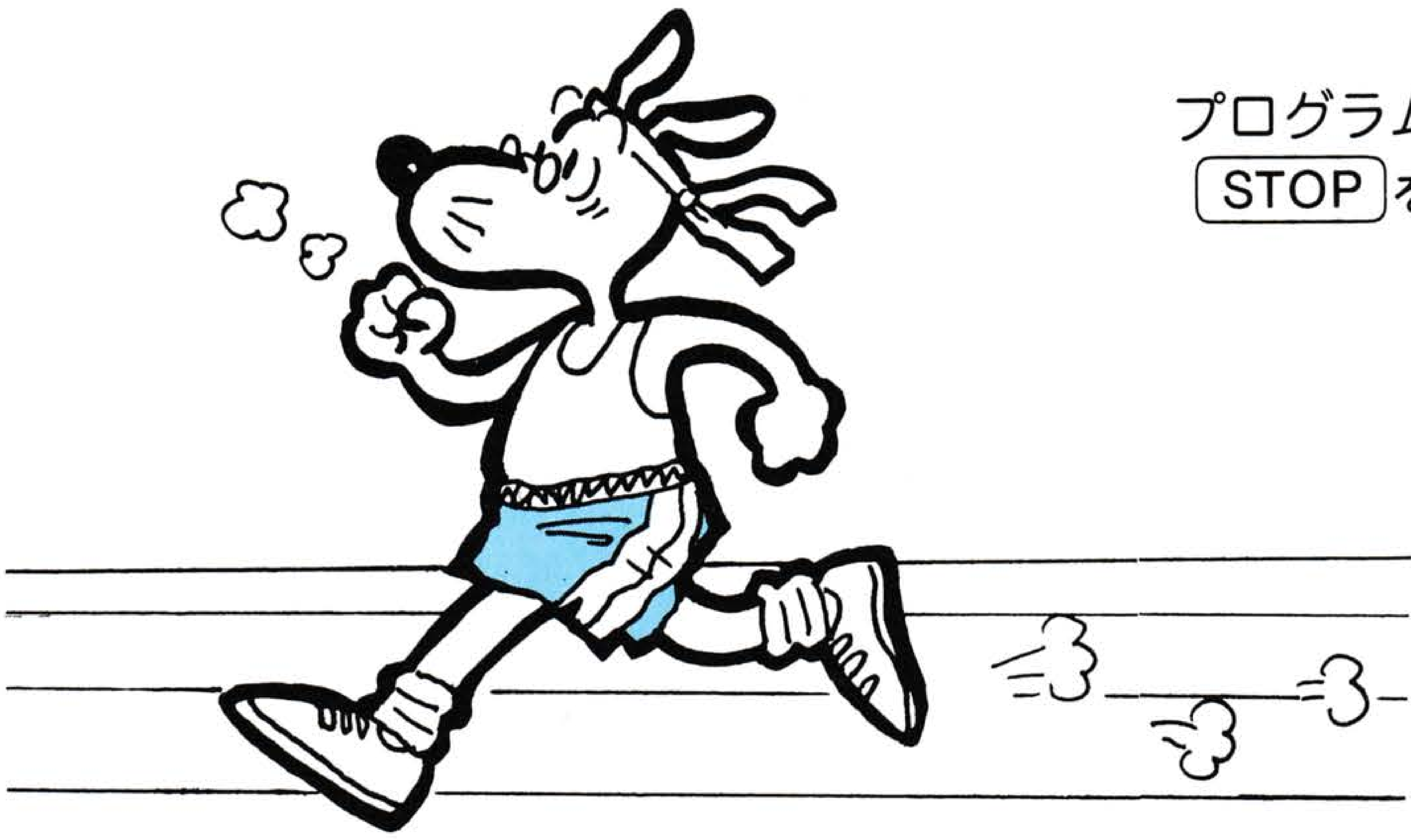
（途中の項目を省略するときには、「，」は省略しないで入力してください）

開始角度，終了角度の値をマイナスの数値にすると、プラスの数値のときと同じ円弧を描いたあと、中心から円弧の両端に直線が描かれ、扇形になります。

・**比率**は、だ円を描くときに入力します。（タテ方向の半径／ヨコ方向の半径）の数値で入力します。数値が１以上ならタテに長いだ円になり、１以下ならヨコに平たいだ円になります。省略したときは、１になります。

スプライトって何だろう

スプライトは、画面に表示されている絵や図形の上に重ねて表示し、速く移動させることができます。また、移動させるときには同じスプライト面の新しい位置にスプライトを表示するだけで前のスプライトが自動的に消える便利な機能をもっています。



プログラムを止めるときは、CTRL を押しながら STOP を押してください。

リストと解説

```
1 'save"ｽﾌﾟﾗｲﾄ.bas"
100 SCREEN 2,1
110 '--- ｽﾌﾟﾗｲﾄ ﾊﾟﾀｰﾝ ﾉ ﾃｲｷﾞ
120 FOR J=0 TO 4
130   B$=""
140   FOR K=1 TO 8
150     READ A$:B$=B$+CHR$(VAL("&H"+A$))
160   NEXT K
170   SPRITE$(J)=B$
180 NEXT J
190 '* ﾊｲｹｲ ｦ ｴｶﾞｸ *
200 COLOR 15,4,7:CLS
210 FOR N=1 TO 10
220   X=N*N+95
230   LINE (0,X)-(256,X),12
240 NEXT N
```

- 100：画面をスクリーン2に、スプライトの種類を1（8×8ドット拡大）に設定
- 120：スプライトパターン5個を定義する
- 130：文字型変数B$の値を初期化する
- 140：8×8ドットパターンのデータを8回読む
- 150：DATA文から読み取ったデータをA$に入れ16進数にして文字列に変換、B$の値に連結する
- 170：B$の値をJ番のスプライトのパターンとして定義する
- 200：画面の初期設定と消去
- 210〜240：左右に直線を10本描く

```
250 '**** ｼﾞｯｺｳ ﾙｰﾁﾝ ****
260 X=120:Y=90:G=0:T=0:PLAY "
S9M400T250"
270 '
280 PUT SPRITE 0,(X,Y),10,T*2
+G:PUT SPRITE 1,(X,Y),13,4
290 NX=50+G*10:PLAY "L16N=NX;
"
300 FOR M=0 TO 300:NEXT
310 '--- ｶｰｿﾙｷｰ
320 S=STICK(0)
330 IF S=3 THEN T=0:GOTO 360
340 IF S=7 THEN T=1:GOTO 360
350 GOTO 310
360 '
370 G=1-G
380 ON T+1 GOTO 390,420
390 '--- T=0
400 X=X+4:IF 240=<X THEN X=24
0
410 GOTO 270
420 '--- T=1
430 X=X-4:IF   X=<0 THEN X=0
440 GOTO 270
450 '--- ｽﾌﾟﾗｲﾄ ﾊﾟﾀｰﾝ ﾃﾞｰﾀ
460 DATA 18,19,66,80,0,A4,42,
2
470 DATA 18,18,20,14,0,18,28,
28
480 DATA 18,98,66,1,0,25,42,4
0
490 DATA 18,18,4,28,0,18,14,1
4
500 DATA 0,0,18,18,18,0,0,0
```

表示位置や状態をあらわす変数を初期設定し、音色を設定する

人物の体のスプライトと衣服のスプライトを重ねて表示する

音を鳴らす

移動スピードを調節するためのタイマー

どのカーソルキーが押されているか調べる

右向きのキーならＴの値を０にして360行へ

左向きのキーならＴの値を１にして360行へ

どちらも押されていなければ、310行へ戻る

姿勢を表すＧの値を反対にする

Ｔに１を加えた値により390または420行へ

ヨコ座標に４を加え、240以上なら240にする

270行へ戻る

ヨコ座標から４を引き、０以下なら０にする

270行へ戻る

人物が右向きで手足を開いたときのパターンのデータ

人物が右向きで手足を閉じたときのパターンのデータ

人物が左向きで手足を開いたときのパターンのデータ

人物が左向きで手足を閉じたときのパターンのデータ

人物の衣服のパターンのデータ

このプログラムを実行させると、画面に人物のスプライトが表示されます。左右のカーソルキーを使って、人物を走らせてみてください。

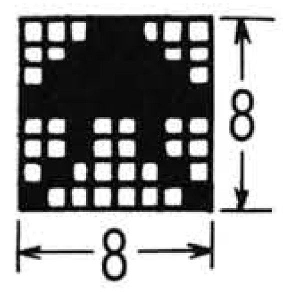
新しい命令の説明

スプライトの種類

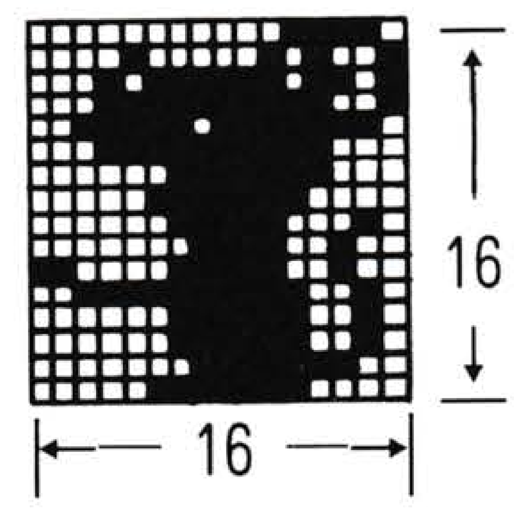
スプライトには、次の 4 種類があります。

スプライトの種類	表示される大きさ	使用できる画面モード	番号の範囲
0	8×8ドット標準	スクリーン1～12	0～255
1	8×8ドット拡大	スクリーン1～12	0～255
2	16×16ドット標準	スクリーン1～12	0～ 63
3	16×16ドット拡大	スクリーン1～12	0～ 63

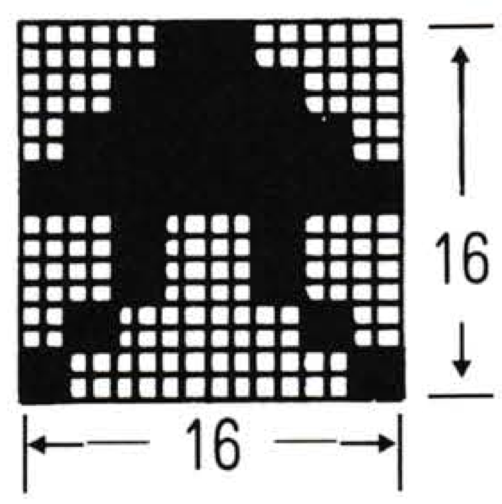
8×8ドット標準
（例：UFO）



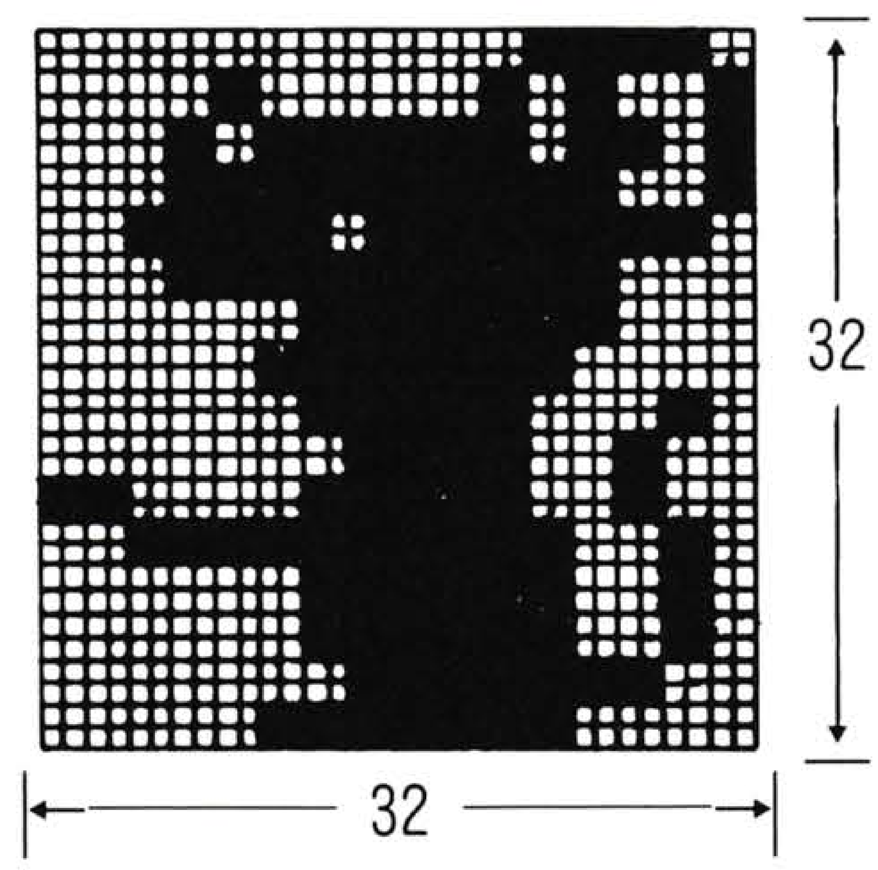
16×16ドット標準
（例：ねずみ）



8×8ドット拡大



16×16ドット
拡大



拡大は標準の大きさをタテ､ヨコとも 2 倍の大きさにしたものです。

SCREEN文によるスプライトの種類の指定

スプライトの種類は、ＳＣＲＥＥＮ文で指定します。

```
100 SCREEN 2, 1
```

画面モード　スプライトの種類

一度スプライトの種類を指定すると、すべてのスプライトが指定した大きさになります。
スプライトの種類を省略したときは、8×8ドット標準のスプライトだけが使用できます。

SPRITE$
(スプライト・ダラー)
(システム変数)

スプライトのパターンを決めることを「スプライトを定義する」と呼びます。

スプライトの定義は、ＳＰＲＩＴＥ$というシステム変数に値を入れることによって行います。

```
170 SPRITE$(J)=B$
```
(J)：スプライト番号　B$：文字列

・システム変数は、ＢＡＳＩＣが特別に使用する変数で、変数名を変えることができません。
・ＳＰＲＩＴＥ$は、文字型変数の１種ですが、表示したいパターンのドットを８ドットずつまとめて１つの文字とし、パターンの種類によって異なる数の文字を１つの文字列としてまとめたものを値にします。

例） ８×８ドット標準の場合

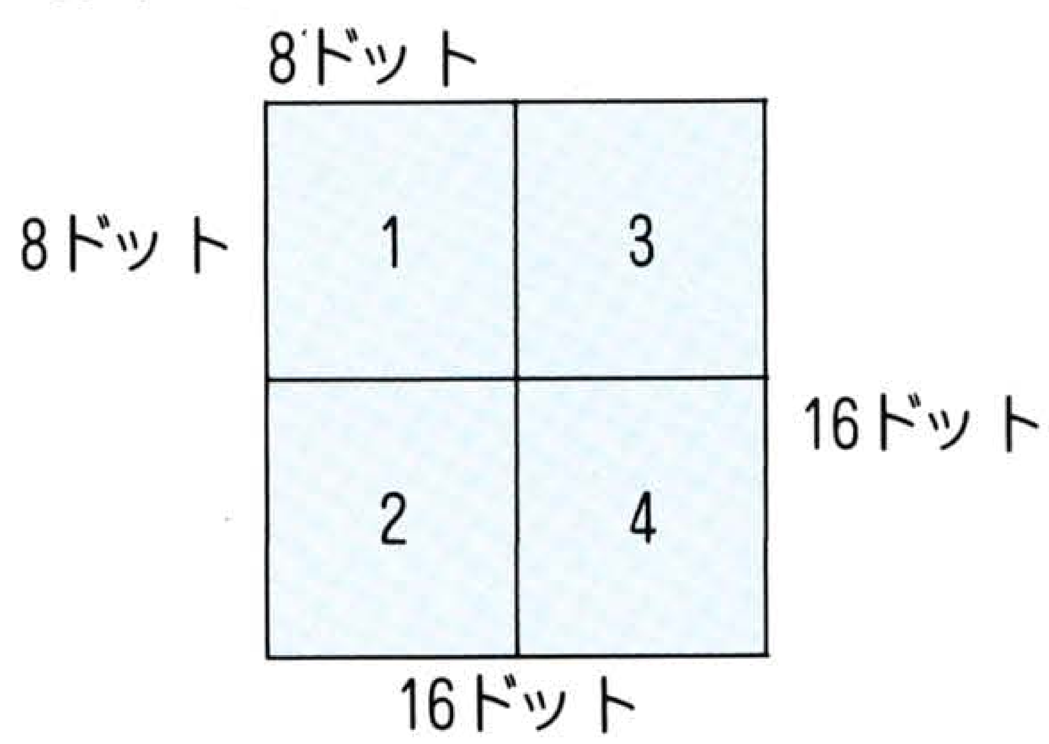
0	0	0	1	1	0	0	0	(&B00011000)	=CHR$(24)
0	0	1	1	1	1	0	0	(&B00111100)	=CHR$(60)
0	1	0	1	1	0	1	0	(&B01011010)	=CHR$(90)
0	1	1	1	1	1	1	0	(&B01111110)	=CHR$(126)
1	1	1	1	1	1	1	1	(&B11111111)	=CHR$(255)
0	0	1	0	0	1	0	0	(&B00100100)	=CHR$(36)
0	1	0	0	0	0	1	0	(&B01000010)	=CHR$(66)
0	1	0	0	0	0	1	0	(&B01000010)	=CHR$(66)

※SPRITE$(0)=CHR$(24)+CHR$(60)+CHR$(90)+CHR$(126)+CHR$(255)+C HR$(36)+CHR$(66)+CHR$(66)

※左の図の１の部分が、指定した色または前景色で画面に表示されます。

※&Bは２進数をあらわします。

・８×８ドット拡大のスプライトは、８×８ドット標準のスプライトと同じ様に定義します。画面に表示されるときには、定義されたパターンの１ドットが画面の４ドット（タテ２ドット×ヨコ２ドット）に拡大されます。
・16×16ドットのスプライトは、８×８ドットのパターン４個に分割し、次の順番で定義します。標準も拡大も定義の方法は同じです。



VAL
(バリュー)
(関数)

VALは、引数としてあたえられた文字列のあらわす数値を得る関数です。

数字でできた文字列を数値に変換するときに使います。

例)

```
A=VAL (B$)
C=VAL ("&HFF")
```

・文字列の先頭の文字が数字または+,ー,&でないときは、VALは値として0に変換します
また、文字列の途中に数字以外の文字があるときは、そこから後ろの文字列は無視します。
ただし、スペース(空白)は無視して変換します。

・このプログラムでは、READ文でデータとして読んだ文字列を16進数に変換するため、文字列の先頭に"&H"を付けてVAL関数の引数にしています。

PUT
SPRITE
(プット
・スプライト)

PUT SPRITEはスプライトパターンを表示する命令です。

```
280 PUT SPRITE 1, (X, Y), 13, 4
```

1:スプライト面番号　(X, Y):位置座標　13:カラーコード　4:スプライトパターン番号

・**スプライト面番号**は、スプライトパターンを表示するスプライト面を指定する番号で、0~31の数値または数値型変数で入力します。1つのスプライト面には1個のスプライトパターンだけを表示できます。2つ以上のスプライトパターンを同時に画面に表示するときは、それぞれ表示するスプライト面を変えてください。同じスプライト面に表示すると、先に表示したスプライトパターンが消えてしまいます。

・スプライト面番号は省略できます。省略したときは、最後に使用したスプライト面を指定したことになります。

・2つ以上のスプライトパターンが重なると、スプライト面番号の小さいスプライトパターンが番号の大きいスプライトパターンの上に重なって表示されます。

・スクリーン1~3では、画面の水平方向に5個以上のスプライトパターンが並ぶと、スプライト面番号の小さいものから順に4個までは表示されますが、5個目からあとのスプライトパターンは消えてしまいます。スクリーン4~12では、8個まで並べて表示できます。この理由で消えたスプライトパターンは、並んでいるスプライトパターンの数が減れば再び表示されます

・**位置座標**は、スプライトパターンを表示する位置を指定する数値です。
指定した位置がスプライトパターンの左上角になります。ヨコ座標は-32~255,タテ座標は-32~191(スクリーン5~8では-32~211)の範囲で指定します。
スクリーン6と7では、絵や図形を表示するときのヨコ座標の範囲が511までになりますが、スプライトの座標は255までを使用します。(画面右端が255です。)スプライトパターンが表示される大きさも変わりません。

・位置座標は省略できます。省略したときは、そのスプライト面に最後に表示したスプライトパターンの位置を指定したことになります。

PUT
SPRITE
のつづき

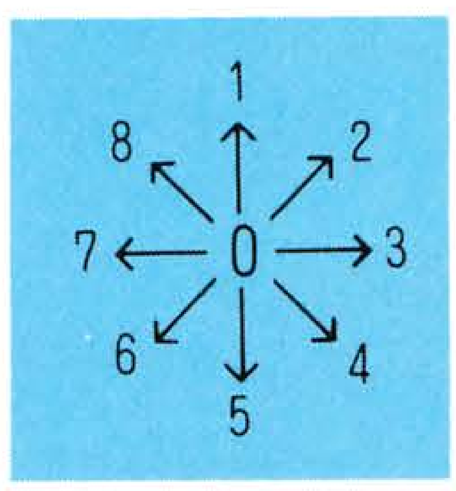
・タテ座標に 208（スクリーン4～8では 216）を指定すると、その面からあとのすべてのスプライト面の表示が消えます。
・タテ座標に 209（スクリーン4～8では 217）を指定すると、そのスプライト面の表示が消えます。
・**カラーコード**は、スプライトパターンを表示する色を番号で指定します。
指定できるカラーコードの範囲は、画面モードによって変わります。（⇨62ページ）
省略したときは、前景色を指定したことになります。
・**スプライトパターン番号**は、ＳＰＲＩＴＥ$変数を使って定義したスプライトパターンを番号で指定します。定義していない番号を指定すると、画面に意味のない模様がスプライトとして表示されることがあります。
・スプライトパターン番号は省略できます。省略したときは、そのスプライト面に最後に表示したスプライトパターンを指定したことになります。

ＳＴＩＣＫ
（スティック）
（関数）

ＳＴＩＣＫは、カーソルキーやジョイスティックまたはジョイパッドがどの方向に押されているかを調べる関数です。

```
320  S=STICK(0)
```
（0）：機器の番号

・調べた結果は、数値で得ます。



・**機器の番号**は、次のように数値で指定します。
0 ‥‥‥‥‥カーソルキーを調べるとき
[↑] が押されていれば１、[→]が押されていれば３、 [↑] と[→]がいっしょに押されていれば２になる
1 ‥‥‥‥‥汎用ポート１に接続したジョイスティックまたはジョイパッドを調べるとき
2 ‥‥‥‥‥汎用ポート２に接続したジョイスティックまたはジョイパッドを調べるとき

第3章
お待たせ！グラフィック

ON GOTO
(オン・ゴートゥー)

ON　GOTOは、指定された変数の値によって次に実行するプログラムの行を変える命令です。

```
380 ON T+1 GOTO 400, 430
```

変数名：T+1　行番号：400　行番号：430

指定された変数の値が1なら左から1番目の行番号へ、2なら2番目の行番号の行へというようにジャンプします。

・**変数名**の部分は、数値や計算式でもかまいません。
・変数の値が1より小さいときや指定された行番号の数より大きいときは、何もしないで次の行に進みます。
・変数の値が整数でないときは、小数部を無視します。

このプログラムでは、状態（右向きか左向きか）をあらわす変数の値により、次にスプライトパターンを表示する位置を決める行を変えています。

《テクニック》

8×8ドットのスプライトパターンの定義方法

8×8ドットのスプライトパターンを定義するときのテクニックとして、下の方法をおすすめします。

```
100 FOR N=1 TO 8
110 READ A$
120 B$=B$+CHR$(VAL("&B"+A$))
130 NEXT N
140 SPRITE$(C)=B$
          :
900 DATA 00011000
910 DATA 00111100
920 DATA 01011010
930 DATA 01111110
940 DATA 11111111
950 DATA 00100100
960 DATA 01000010
970 DATA 01000010
```

・ＤＡＴＡ文を0と1だけで表記する（900行～）
・読み取ったデータを2進数として数値に変換してから文字列に変換する（120行）

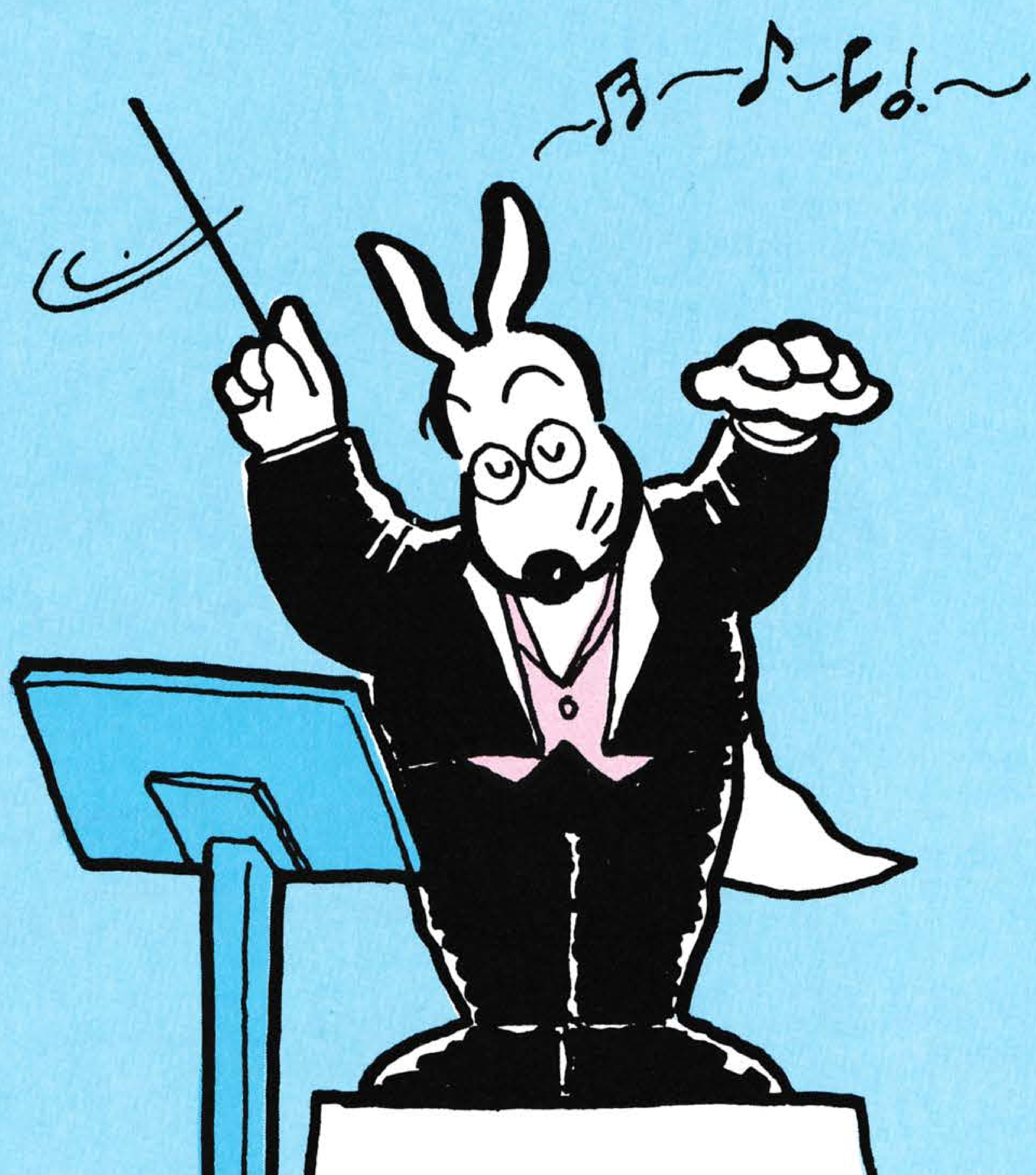
この方法なら、ＤＡＴＡ文を入力するときにスプライトパターンを目で確かめながら作ることができます。

第4章

音楽がない生活なんて考えられない！という人のパソコンライフのために

BASICでMUSIC!

まず♪ドレミから ………………………………… 84
（ＰＬＡＹ文、ＭＭＬ、サンプルプログラム）
ＦＭ音源で演奏してみよう ……………………… 88
（演奏のサンプルプログラム）
キーボードで音楽を ……………………………… 92
（ＰＬＡＹ関数、サンプルプログラム）
ＰＣＭ音源で録音・再生してみよう …………… 95
（サンプルプログラム）

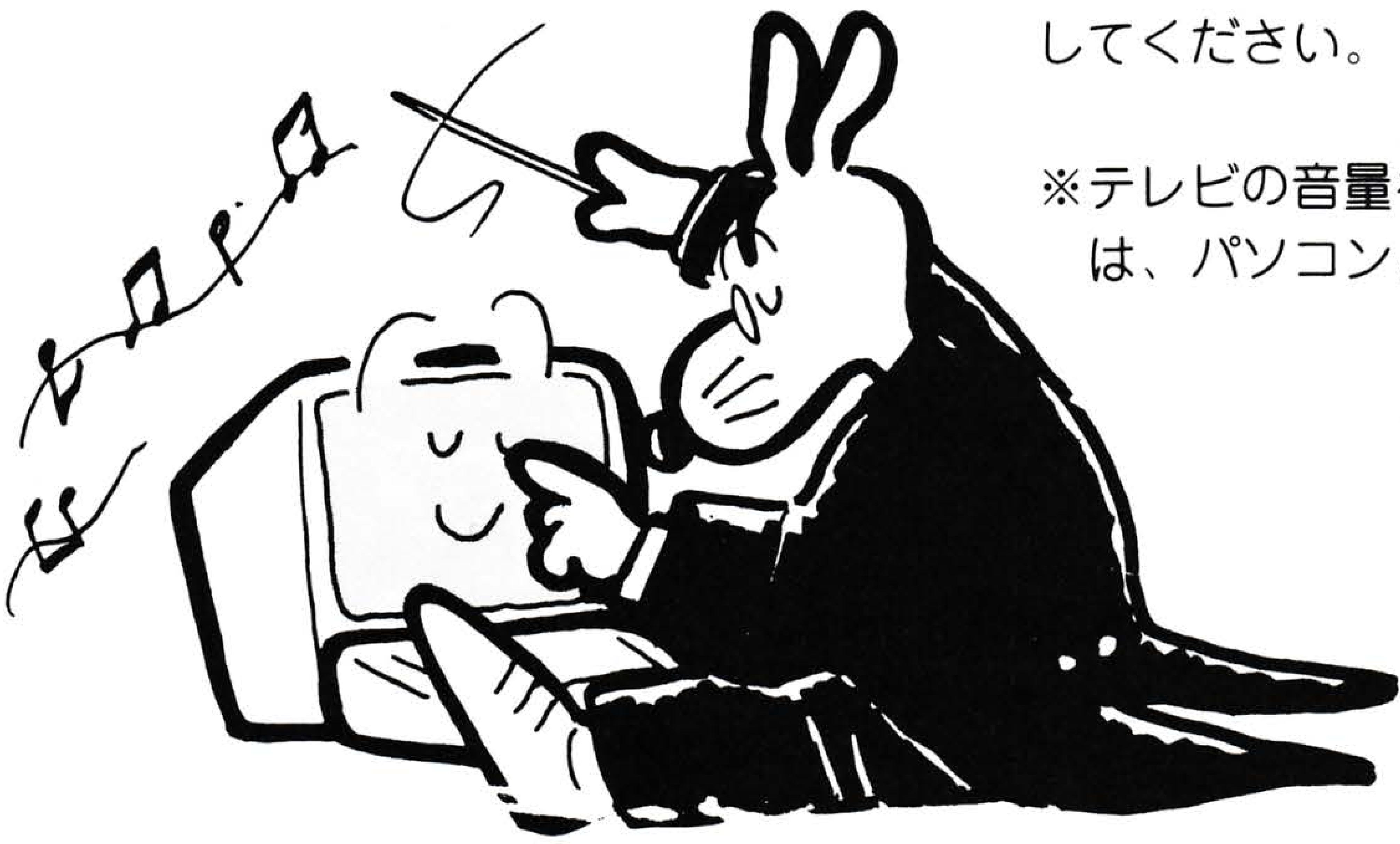


まず♪ドレミから

パソコンでゲームをしたことのある人は、ゲームがスタートするときなどにきれいな音楽が流れるのを聞いたことがあるでしょう。あなたのＭＳＸ turbo Ｒパソコンにも、音楽を演奏する機能がついています。ＢＡＳＩＣで音楽を演奏させてみましょう。

演奏が終わればプログラムは止まりますが、途中で止めたいときは CTRL を押しながら STOP を押してください。

※テレビの音量を大きくしても何も聞こえないときは、パソコンとテレビの接続を確かめてください。



リストと解説

```
1 'save"ドレミ.bas"
10 FOR N=0 TO 1
20   PLAY "V15O4CDEFGABO5C8.R
16"
30   PLAY "CO4BAGFEDC8.R16"
40 NEXT N
```

――″ドレミファソラシド″を繰り返し演奏するプログラムです。

```
1 'save"キリツレイ.bas"
10 PLAY"T90V15O4L2","T90V15O4
L2","T90V15O4L2"
20 PLAY "EDE","GGG","O5CO4BO5
C"
```

――学校で聞いた覚えがあるでしょう？和音を鳴らすプログラムです。

```
1 'save"エリーゼ.BAS"
10 PLAY "V15T120L8O5EE-EE-EO4
BO5DCO4A4."
20 PLAY "R8CEAB4.R8EG#BO5C4.R
8"
```

――半音を使った演奏のプログラムです。

新しい命令の説明

MSX turbo
Rパソコンの
音楽機能

MSX turbo Rパソコンは、「PSG」・「FM音源」・「PCM音源」という3種類の音源（音を発生するためのLSI）をもっています。

これらの音源には、それぞれに特長があります。
- 「PSG」は最も基本的な音源で、いちばん手軽に音を発生することができます。
- 「FM音源」は本体に内蔵されている様々な楽器や効果音のデータを利用して、美しい音色を発生することができます。
- 「PCM音源」はテープレコーダのように、音声を録音・再生するために使用します。

ここでは、まず「PSG」を使っています。

PSGのボイス
チャンネル

PSGは、発生する音の高さや大きさを調整する回路を3組もっており、同時に3つの音を発生することができます。

そこでBASICでは、この3組の回路を「ボイスチャンネル1～3」として、別々に操作できるようになっています。

PLAY
（プレイ）

音楽を演奏したり、演奏についての各種の設定を行う命令です。

```
20 PLAY ″EDE″, ″GGG″, ″O5CO4BO5C″
       ″文字列″ ″文字列″   ″文字列″
```

- あらかじめ使用する音源を指定せずにPLAY文を使用すると、PSGが指定されます。
- ″文字列″は、**「MML」**（ミュージック・マクロ・ランゲージ）と呼ばれる文字の前後を、「″」でくくったものです。
  MMLは一定の英数字と記号の集まりで、音楽の演奏についての細かい命令の役割をします。
  MMLについては、185ページの一覧表にのっています。
- ″文字列″は、MMLを値とする文字型変数や式でもかまいません。
- PSGでは、1つのPLAY文で3つまでの″文字列″を入力することができます。
  左側の″文字列″から順にボイスチャンネル1、ボイスチャンネル2、ボイスチャンネル3についての命令とみなされます。ボイスチャンネル2や3で演奏するときは、ボイスチャンネル1を使わないときでも、前に「″″」（ヌル：値がない文字列）を入力することが必要です。
  例）　**PLAY ″″, ″″, ″CDE″**
- 1つのPLAY文に2つ以上の″文字列″が入力されているときは、各文字列に書かれたMMLが同時に実行（演奏）されます。
  例）　PLAY ″C″, ″E″, ″G″　　ドミソの和音になります。
  　　　PLAY ″CC″, ″EF″, ″GA″　ドミソとドファラの2和音を演奏します。

サンプルプログラムで使用したMML

・O……………そのあと演奏する音のオクターブの高さを後ろの数字で指定します。
数字の範囲は1～8で、スタート時は4になっています。この指定は、次に「O」を指定するときまで有効です。つまり、プログラムの実行を終わっても設定が残ったままですので、そのあとで別のプログラムを実行したときにも影響します。

・L……………そのあと演奏する音の長さを後ろの数字で指定します。
数字の範囲は1～64で、「何分音符」かを指定します。スタート時は4です。
この指定は、次にLを指定するまで有効です。

・A～G………発生する音の高さをそのオクターブの中での音程で指定します。
C（ド）D（レ）E（ミ）F（ファ）G（ソ）A（ラ）B（シ）
後ろに数字を付けると、直前の音の長さを指定したことになります。

・「.」………付点。直前の音の長さを 1.5倍します。

・R……………休符。後ろの数字で「何分休符」かを指定します。
休符はLによる音の長さの指定に影響されないので注意してください。
数字を付けないときは4分休符になります。

・T……………テンポの指定。後ろの数字で1分間に4分音符で何拍するテンポかを指定します。
数字の範囲は 32～255で、スタート時は 120です。
この指定は、次にTを指定するまで有効です。

・V……………音量。後ろの数字で音の大きさを指定します。数字の範囲は0～15で、スタート時は8です。この指定は、次にVを指定するまで有効です。

・ー……………♭（フラット）。直前の音の高さを半音下げます。

・♯……………シャープ。直前の音の高さを半音上げます。＋を代用できます。

MML使用時の注意

・各″文字列″のMMLは、そのボイスチャンネルだけに有効です。たとえば、2つ以上のボイスチャンネルを使って演奏する曲のテンポを変えるときには、同じＰＬＡＹ文の中のすべての″文字列″でTを指定し直すことが必要です。

・O，L，T，Vの指定は、そのボイスチャンネルで次に指定し直すまで有効です。
したがって、これらの設定を変更するプログラムを実行させたあとで別の演奏プログラムを実行させるときは、最初にこれらを指定し直してください。

・Oは数字の0と特にまちがいやすいので注意して入力してください。

参考プログラム

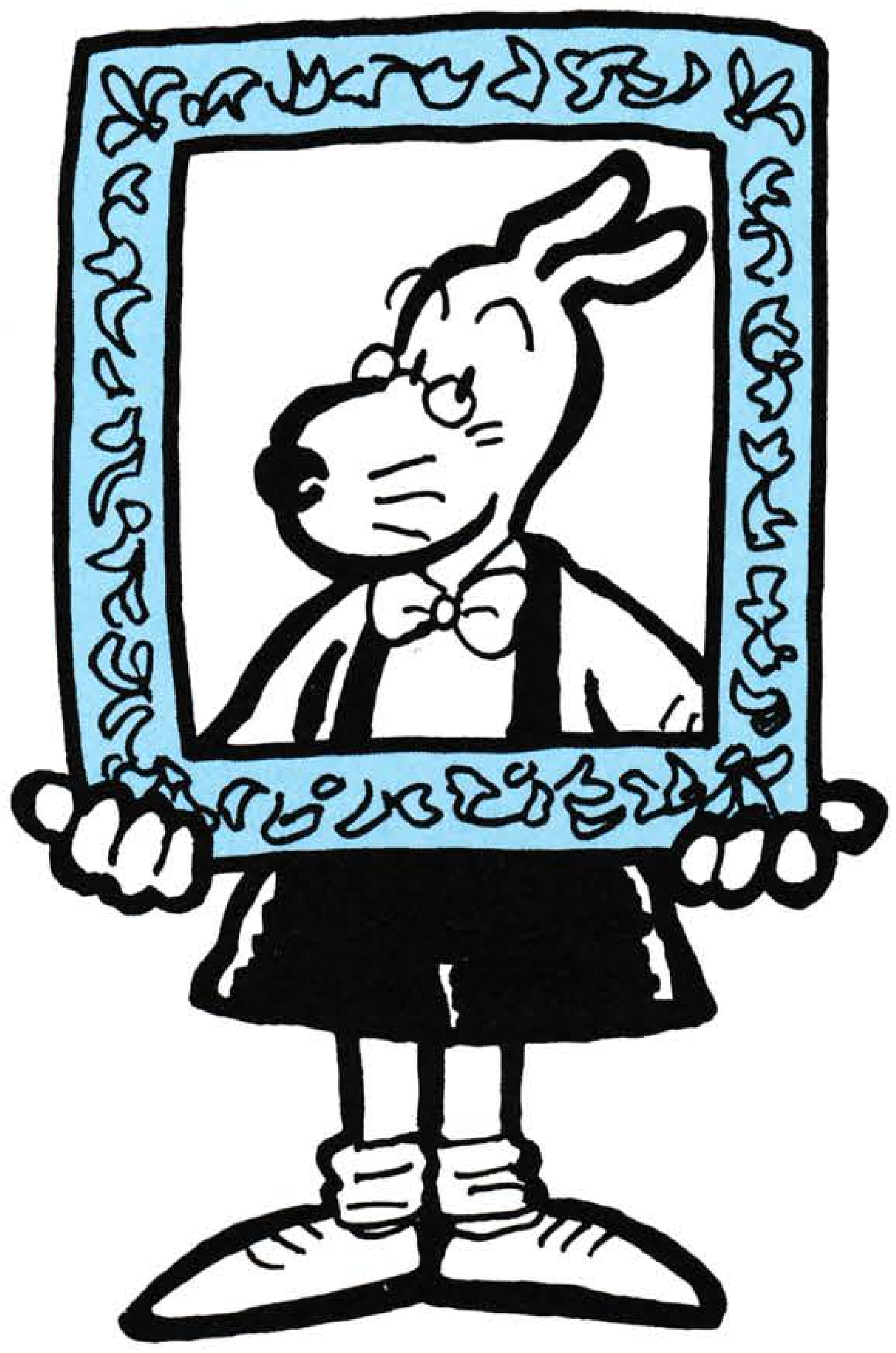
```
1 'save"ﾊﾞｯﾊ.BAS"
100 PLAY "T160L6V15","T160L2V
12"
110 PLAY "R6O6GABO7DCCED","O4
GO5GE"
120 PLAY "DGF#GDO6BGAB","O4BO
5EO4E"
130 PLAY "O7CDEDCO6ABG","ABO5
C"
140 PLAY "F#GADF#AO7CO6BA","O
5DF#D"
150 PLAY "BGABO7DCCED","GGC"
160 PLAY "DGF#GDO6BGAB","O4BO
5ED"
170 PLAY "EO7DCO6BAGDGF#","CC
#D"
180 PLAY "G2","G"
```

```
1 'save"ﾊﾅ.bas"
10 PLAY "T60V15L16","T60V12L1
6"
20 PLAY "O5D8.DG8.GAGF#ED4","
O5D8.DO4B8.BCO5EDO4CB4"
30 PLAY "O4BO5DGAB8O6D8O5A4.R
8","O4B8BO5DG8G8F#4.R8"
40 PLAY "O5D8.DG8.GAGF#ED4","
O5D8.DO4B8.BCO5EDO4CB4"
50 PLAY "O5A8.AA8B8G4.R8","O4
A8.O5F#F#EDCO4B4.R8"
```

２本とも２パートの同時演奏を行うプログラムです。どちらも今まで説明したことだけを使って作られていますので、演奏プログラムを作るときの参考にしてください。

FM音源で演奏してみよう

ＦＭ音源を使って、ムソルグスキーの「展覧会の絵」の一部を演奏してみましょう。



このプログラムは演奏が終わると止まります。

※途中で止めたいときは、［CTRL］を押しながら［STOP］を押してください。

リストと解説

```
1 'save"FMﾃﾝﾗﾝｶｲ.bas"
100 CALL MUSIC
110 CALL VOICE(@0,@2,@3)
120 READ A$,B$,C$
130 IF A$="END" THEN END
140 PLAY #2,A$,B$,C$
150 GOTO 120
160 '* えんそう デ-タ *
170 DATA V13L4,V10L4,V10L4
180 '
190 DATA L4O4GFB-O5C8F8D,,
200 '
210 DATA C8F8DO4B-O5CO4GF,,
220 '
230 DATA O4G FB-O5C8F8D
240 DATA O4D CD   A   A
250 DATA O3B-AB-O4C   F
```

- 100：ＦＭ音源の使用を宣言する（準備させる）
- 110：各チャンネル別にＦＭ音源の音色を割り当てる
- 120：１７０行からのＤＡＴＡ文からデータを読む
- 130：データが「ＥＮＤ」なら、実行を終了する
- 140：読み取ったデータにしたがって演奏する
- 150：１２０行に戻る
- 170：各ボイスチャンネルの音量と音の長さのデータ
- 190・210：ボイスチャンネル１だけの演奏データ（他のチャンネルはまだ演奏しない）
- 230～250：３つのチャンネルで演奏するデータ（３行でひと組み）

```
260 '
270 DATA O4C8F8DO4B-O5CO4GF
280 DATA O4A   B- G   G  CC
290 DATA O4C   C  D   EO3GA
300 '
310 DATA O4B-O5C8F8DC8F8DO4B-
320 DATA O4F   A8R8B-A8R8B- F
330 DATA O3D   C O2B-A GO3G
340 '
350 DATA O5CO4GF  G FB-
360 DATA O4G  EC  D CD
370 DATA O3C  EFO2B-AG
380 '
390 DATA O5C8F8D O4B-O5E -CO4
B-
400 DATA O4A8R8B- G O5CO4A
F
410 DATA O2F  B-O3G  C    F
B-
420 '
430 DATA END,,
```

※MMLの間の空白は、リストを見やすくするために挿入したもので、演奏時には無視されます。(入力しなくてもかまいません。)

新しい命令の説明

CALL MUSIC (コール・ミュージック)

FM音源を使える状態にするための命令です。

```
100 CALL MUSIC
```

・FM音源やPCM音源は、それらの機能を使用することをパソコンにあらかじめ教えておかなければ使用できません。このような役目をする命令は、できるだけプログラムの先頭に近い部分に置きます。特に、その機能に関する他の命令より先に、必ず実行させてください。

・CALL MUSICには、FM音源のボイスチャンネルを割り当てる機能もあります。

例) CALL MUSIC(1, 0, 3, 1, 2)

リズム音使用　　ボイスチャンネルの割り当て

意味なし

FM音源には、全部で9つのボイスチャンネルがあります。各ボイスチャンネルは、それぞれ別の音を発生できますが、内蔵のリズム音を使用するには3チャンネル必要なため、メロディの演奏や効果音の発生に使用できるのは6チャンネルまでになります。

CALL MUSIC (つづき)

・リズム音の使用は、かっこ内の最初の数字で指定します。（1：使用、0：不使用）
かっこ内の2番目の数字は、常に「0」にしてください。
かっこ内の3番目から後の数字は、PLAY文の各″文字列″に対するボイスチャンネルの割り当て数を表します。上の例では、PLAY文の1番目の″文字列″に3チャンネル、2番目の″文字列″に1チャンネル、3番目の″文字列″に2チャンネルを割り当てています。
FM音源では、PLAY文の″文字列″の数は4つ以上でもかまいません。
100行のように、かっこ全体を省略したときは、次の設定をしたものとみなされます。

CALL MUSIC（1，0，1，1，1）

CALL VOICE (コール・ボイス)

FM音源の各ボイスチャンネルに音色を設定する命令です。

```
110 CALL VOICE (@0, @2, @3)
```

チャンネル1の音色番号、2の音色番号、3の音色番号

音色は「音色データ一覧表」（⇨186ページ）から選び、「@」（アットマーク）と音色番号で指定します。指定を省略したチャンネルには、「@0」（ピアノ1）の音色が設定されます。
音色データ一覧表で＊マークのない音色や配列変数は、1つのVOICE文で2つ以上設定できません。また、CALL MUSIC文で設定した演奏用のチャンネル数を越えて音色を設定することはできません。

PLAY# (プレイ・シャープ)

FM音源を使って音楽の演奏や各種の設定をする命令です。
PSGを一緒に操作することもできます。

```
140 PLAY #2, A$, B$, C$
```

″文字列″ ″文字列″ ″文字列″

「#」（シャープ）の後の数字は、2または3にしてください。どちらでも結果は同じです。
「#」の後の数字を0にすると、PSGのPLAY文と同じになりPSGしか操作できません。
「#」の後の数字を1にすると、エラーになります。
″文字列″には、140行のように、MMLを値とする文字変数が使えます。
″文字列″は、次の順に並べます。

例） PLAY#2，″FM音源用MML1″，″FM音源用MML2″，……″FM音源用MMLn″，″リズム音用MML″，″PSG用MML1″，″PSG用MML2″，″PSG用MML3″

・″FM音源用MML″は、CALL MUSIC文で設定したメロディや効果音の数だけ設定することが必要です。演奏しない″文字列″は「″″」（ヌル）にしてください。
・″リズム音用MML″は、CALL MUSIC文でリズム音を使用するように設定したときのみ入力します。（リズム音を使用しない設定のときは、「″″」にしないで「,」ごと省略してください。）

PLAY#
(つづき)

・"ＰＳＧ用ＭＭＬ"は、ＰＳＧ用ボイスチャンネル１～３の順に入力します。途中のＰＳＧ用ボイスチャンネルに演奏させないときは、「""」にしてください。
・各ＭＭＬについては、185ページの一覧表を参照してください。

FM音源用拡張BASIC「MSX-MUSIC」

ここで紹介した命令は、いずれも「ＭＳＸ－ＭＵＳＩＣ」と呼ばれるＦＭ音源用拡張ＢＡＳＩＣの一部です。
従来のＭＳＸ２＋ パソコンなどでは、「ＭＳＸ－ＭＵＳＩＣ」は内蔵されていない場合もある機能（オプション）でした。ＭＳＸ turbo Ｒパソコンは、初めて「ＭＳＸ－ＭＵＳＩＣ」を標準装備したものです。
ですから「ＭＳＸ－ＭＵＳＩＣ」には、すでにたくさんの市販ソフトが対応しています。また、楽しい応用プログラムが掲載されている解説書なども出版されていますので、一度、お店で調べてみてみてください。
「ＭＳＸ－ＭＵＳＩＣ」に対応している市販ソフトやハードには、次のマークが表示されています。

キーボードで音楽を

**パソコンのキーボードを使って音楽を演奏できるようにするプログラムです。
これなら、演奏プログラムを作るのがメンドウだという人もＯＫですね？**

※プログラムを止めたいときは、[CTRL]を押しながら[STOP]を押してください。



リストと解説

```
1 'save"ｷｰﾎﾞｰﾄﾞ.bas"
10 CALL MUSIC
20 CALL VOICE(@0)
100 SCREEN 1:WIDTH 32:CLS
110 PLAY #2,"T120":VV=15:LL=8
:GOSUB 410:GOSUB 430
120 '
130 PLAY #2,"XA$;":A$=""
140 '
150 K$=INPUT$(1)
160 IF K$=CHR$(&H1E) AND   VV
<15 THEN VV=VV+1:GOSUB 410
170 IF K$=CHR$(&H1F) AND 0<VV
    THEN VV=VV-1:GOSUB 410
180 IF K$=CHR$(&H1C) AND 1<LL
    THEN LL=LL/2:GOSUB 430
190 IF K$=CHR$(&H1D) AND   LL
<64 THEN LL=LL*2:GOSUB 430
200 IF PLAY(0) THEN 140
```

- 10：ＦＭ音源の使用を宣言する（準備させる）
- 20：チャンネル１を「ピアノ１」の音色にする
- 100：画面の初期設定
- 110：演奏のテンポと音量と音の長さを指定する変数の値を初期設定し、 410行から始まるサブルーチン、430行から始まるサブルーチンにジャンプする
- 130：Ａ$の値のＭＭＬにより音を鳴らす
- 150：押されているキーを調べる
- 160：[↑] が押されていて、変数ＶＶの値が15未満なら、ＶＶの値を１増やして410行へ
- 170：[↓] が押されていて、変数ＶＶの値が０より大きければ、ＶＶの値を１減らして410行へ
- 180：[→] が押されていて、変数ＬＬの値が１より大きければ、ＬＬの値を半分にして430行へ
- 190：[←] が押されていて、変数ＬＬの値が64未満なら、ＬＬの値を２倍にして430行へ
- 200：音が鳴っている途中なら、140行へ

```
210 IF K$="z" OR K$="Z" THEN
A$="O4A"
220 IF K$="x" OR K$="X" THEN
A$="O4B"
230 IF K$="c" OR K$="C" THEN
A$="O5C"
240 IF K$="v" OR K$="V" THEN
A$="O5D"
250 IF K$="b" OR K$="B" THEN
A$="O5E"
260 IF K$="n" OR K$="N" THEN
A$="O5F"
270 IF K$="m" OR K$="M" THEN
A$="O5G"
280 IF K$="," THEN A$="O5A"
290 IF K$="." THEN A$="O5B"
300 IF K$="/" THEN A$="O6C"
310 IF K$="a" OR K$="A" THEN
A$="O4G#"
320 IF K$="s" OR K$="S" THEN
A$="O4B-"
330 IF K$="f" OR K$="F" THEN
A$="O5C#"
340 IF K$="g" OR K$="G" THEN
A$="O5E-"
350 IF K$="j" OR K$="J" THEN
A$="O5F#"
360 IF K$="k" OR K$="K" THEN
A$="O5G#"
370 IF K$="l" OR K$="L" THEN
A$="O5B-"
380 IF K$=":" THEN A$="O6C#"
390 IF K$="]" THEN A$="O6E-"
400 GOTO 120
```

キーボードの下から１段目のキーにピアノの白鍵にあたる音をそれぞれ割り当て、押されたキーの高さの音をＡ$に入れる

２段目のキーに黒鍵の音を割り当てる

120行に戻る

```
410 '
420 PLAY #2,"V=VV;":LOCATE  5
,10:PRINT "V=";VV:RETURN
430 '
440 PLAY #2,"L=LL;":LOCATE 15
,10:PRINT "L=";LL:RETURN
```

（以下はサブルーチン）

420〜：音量を変数ＶＶの値に設定し、画面に表示してメインルーチンに復帰する

440〜：音の長さを変数ＬＬの値に設定し、画面に表示してメインルーチンに復帰する

新しい命令の説明

ＰＬＡＹ（関数）

指定されたボイスチャンネルが演奏中かどうかを調べる関数です。

```
A=PLAY (0)
```

変数名　　チャンネル番号

・チャンネル番号は０〜３の数値で指定します。各数値は下の意味になります。

　０……ボイスチャンネル１〜３のどれか
　１……ボイスチャンネル１
　２……ボイスチャンネル２
　３……ボイスチャンネル３

・調べた結果は、演奏中ならー１、演奏中でなければ０になります。

・ここでいう「演奏中」とは、ＰＬＡＹ文を実行中かどうかを意味します。
　したがって、休符や各種の設定を実行中で実際には音が鳴っていなくても「演奏中」になります。

新しく出てきたＭＭＬ

＝変数名；……ＭＭＬの後ろに付ける数字を変数で指定するときに使います。

```
420  PLAY  #2,"V=VV;"
```

ＭＭＬ　変数名

・変数名は数値型変数であることが必要です。

Ｘ変数名；……入力するＭＭＬを変数で指定するときに使います。

```
130  PLAY  #2,"XA$;"
```

変数名

・変数名は文字型変数であることが必要です。

PCM音源で録音・再生してみよう

ＭＳＸ turbo Ｒパソコンのｐｃｍ音源は、テープレコーダのように、音声の録音・再生ができます。パソコンにしゃべらせて、友達を驚かせましょう。

リストと解説

```
1 'save"ﾛｸｵﾝ.bas"
100 CLEAR1000,&HB000
110 SCREEN 1:WIDTH 32:CLS
120 B$="REC!":GOSUB 180
130 CALL PCMREC(@&HB000,&HDFF
F,1,32,1)
140 B$="PLAY":GOSUB 180
150 CALL PCMPLAY(@&HB000,&HDF
FF,1)
160 END
170 '
180 A$="ﾅﾆｶ ｷｰ ｦ ｵｼﾃｸﾀﾞｻｲ"
190 CLS:LOCATE 5,10:PRINT A$
200 K$=INKEY$
210 IF K$="" THEN 200
220 LOCATE 5,12:PRINT B$:RETU
RN
```

- 100：録音したデータを納める領域を確保する
- 110：画面の初期設定
- 120：最初のメッセージを表示し、何かキーが押されるまで待つ
- 130：ＰＣＭ録音をする
- 140：次のメッセージを表示し、何かキーが押されるまで待つ
- 150：ＰＣＭ再生をする

新しい命令の説明

CALL PCMREC（コール・ピーシーエム・レック）

ＰＣＭ録音をする命令です。

```
130 CALL PCMREC(@&HB000,
        &HDFFF,1,32,1)
```

（&HB000：開始番地、&HDFFF：終了番地、1：細かさ、32：レベル、1：圧縮指定）

ＰＣＭ（ピーシーエム：Pulse Code Moduration）とは､音声を微小な単位時間ごとに測定（サンプリング）して数値化したデータにより録音や再生をすることです。

・録音したデータは本体のメモリに記憶されます。上の例では、１００行のＣＬＥＡＲ命令でメインメモリの中に作った記憶用領域にデータを保存するため、その領域の「開始番地」と「終了番地」を指定しています。（⇨ 196ページ「メモリマップ（ＢＡＳＩＣ使用時）」）

・「細かさ」はサンプリングの単位時間の長さを表し、０～３の数値を入力します。数字が大きくなるほど測定の間隔が長くなり、データは粗くなりますが、その分録音に必要なメモリが少なくなり、録音できる時間が長くなります。このプログラムでは、約２秒間録音できます。

・録音は、設定された音量を越える音声がマイクから入力されたときに、自動的に始まります。「レベル」は自動録音を開始する音量を表す数値で、０～１２７の数値を入力します。

CALL
PCMREC
(つづき)

・サンプリングした音声データには多くの無音データが含まれます。「圧縮指定」は、この無音データの部分を詰めた形式で記録（圧縮記録）することにより、録音に必要なメモリを省略するかどうかの指定を行うものです。０または省略のときは圧縮せず、１のときは圧縮します。圧縮記録すると再生したときの音質が少し悪くなりますが、録音できる時間は長くなります。（無音データの量は音声により異なるため、圧縮したときの録音時間は、録音内容により変わります。）

CALL
PCMPLAY
(コール・ピーシーエム・プレイ)

ＰＣＭ再生をする命令です。

```
150 CALL PCMPLAY(@&HB000,
    &HDFFF, 1)
```

（&HB000＝開始番地、&HDFFF＝終了番地、1＝細かさ）

・「開始番地」と「終了番地」は録音したデータの記憶領域を指定します。省略はできません。
・「細かさ」は０～３の数値を入力します。録音したときの「細かさ」と同じ数値を入力しないと、正常な音声で再生できません。

おまけのプログラム

```
1 'save"VRAMﾛｸｵﾝ.bas"
100 SCREEN 8
110 OPEN"GRP:" AS #1
              :
130 CALL PCMREC(@65536!,13107
1!,1,32,1,S)
              :
150 CALL PCMPLAY(@65536!,1310
71!,1,S)
              :
              :
180 SCREEN 8:A$="ﾅﾆｶ ｷｰ ｦ ｵｼﾃ
ｸﾀﾞｻｲ"
190 PRESET(50,100):PRINT #1,A
$
              :
              :
220 PRESET(50,120):PRINT #1,B
$:RETURN
```

録音できる時間を長くするため、ＶＲＡＭに音声データを記憶させるようプログラムを改造してみましょう。左のリストに書かれている行だけを修正し、その他の行は今のままにしておいてください。これで約８秒間録音できます。

・ＶＲＡＭに音声データを記憶させる場合、ＣＡＬＬ　ＰＣＭＲＥＣ命令とＣＡＬＬ　ＰＣＭＰＬＡＹ命令は次の形式で書きます。
  ＣＡＬＬ　ＰＣＭＲＥＣ
  （@開始番地，終了番地，
  細かさ，レベル，圧縮指定，Ｓ）
  ＣＡＬＬ　ＰＣＭＰＬＡＹ
  （@開始番地，終了番地，細かさ，Ｓ）
・ＶＲＡＭに音声データを記憶させる場合、スクリーンモードを４以上にしてください。
・見なれない命令が含まれていますが、興味があるかたは181ページを参照してください。

メモ：ＶＲＡＭの番地を指定するときは、16進数ではなく、10進数で指定してください。

第5章

漢字を使いにくいパソコンなんて・・・・という人のための正しい解決方法

かんたん漢字モード!

漢字モードのスタート …………………………… 98
（漢字モードって何？、漢字モードのスタートのしかた）
漢字の入力のしかた …………………………… 100
（全角文字と半角文字、連文節変換）
漢字モードの種類 …………………………… 111
漢字モードの画面 …………………………… 112
漢字で住所氏名を表示しよう………………… 114
（サンプルプログラム）
漢字グラフィック画面で文字表示 ………………… 116
（サンプルプログラム）
「入門」のおわりに …………………………… 118

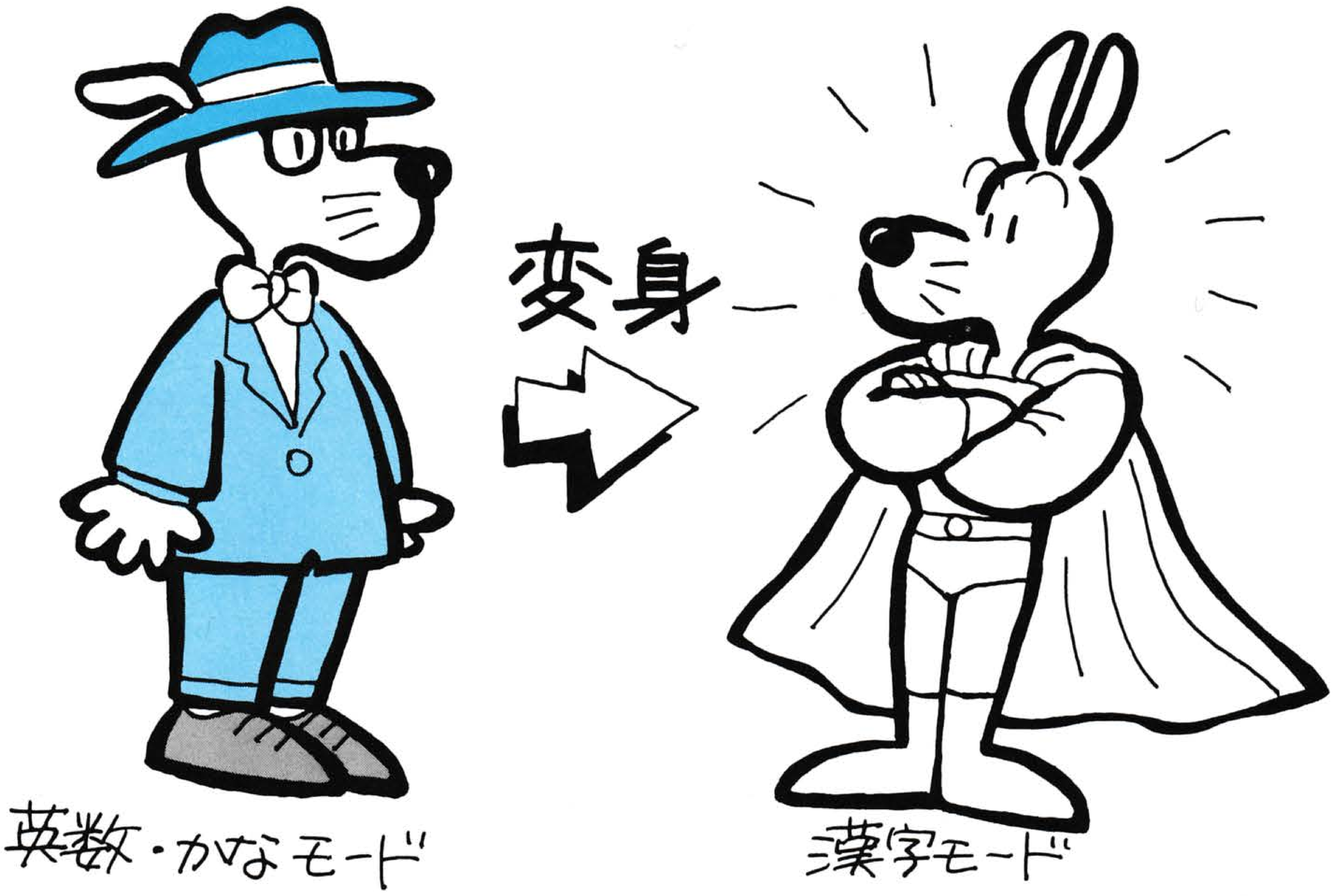


漢字モードのスタート

これからのページでは、漢字モードについて説明します。

漢字モードって何?

ＭＳＸ turbo ＲのＢＡＳＩＣでは、これまでのＭＳＸやＭＳＸ２ＢＡＳＩＣにくらべ、ずっとかんたんに漢字が使えるようになっています。ただ、これまでのＢＡＳＩＣと同じやりかたでも漢字をあつかえるようにしているため、電源スイッチを入れたときから新しい漢字機能を使うことはできません。まずＢＡＳＩＣに「頭の切り替え」をさせることが必要なのです。
この「頭の切り替え」をした状態を**「漢字モード」**と呼びます。



それに対して、これまで説明してきたＢＡＳＩＣがスタートしたときのままの状態を**「英数・かなモード」**と呼びます。

漢字モードの特長

・キーボードから「かな」で読みを入力し、画面上で変換することにより、漢字を直接入力できる
・プログラムの中でＰＲＩＮＴ文やＩＮＰＵＴ文の文字列（文字定数）やＲＥＭ文に漢字を使える
・漢字の文字列を操作するための新しい命令が使える

これらが漢字モードの主な特長です。

漢字モードのスタートのしかたと終了のしかた

```
CALL KANJI
```

ＢＡＳＩＣを英数・かなモードから漢字モードに切り替えるには、上の命令を実行させます。

ダイレクトモードでは画面の表示が消え、左上にそれまでより大きな文字で「Ｏｋ」と表示されます。これでＢＡＳＩＣが漢字モードに切り替わっています。
この命令を**「ＣＡＬＬ　ＫＡＮＪＩ（コール・カンジ）命令」**と呼びます。

```
CALL ANK
```

漢字モードから英数・かなモードに切り替えるには、上の命令を実行させます。

ダイレクトモードでは再び画面の表示が消え、左上に元の大きさの文字で「Ｏｋ」と表示されます。これで英数・かなモードに切り替わっています。
この命令を**「ＣＡＬＬ　ＡＮＫ（コール・アンク）命令」**と呼びます。

漢字モードでの命令などの入力のしかた

まずＢＡＳＩＣを漢字モードに切り替え、英数・かなモードと同じように、キーボードの文字キーを押してみましょう。

英数・かなモードと同じように、押したキーの文字が画面に表示されますね。
ただ、文字のタテの大きさが英数・かなモードより大きくなりますので、文字によってはかなり形が違って見えます。まちがえないように注意してください。

この大きさの文字を**「半角文字」**と呼びます。半角文字は英文字、数字、記号、カタカナ、グラフィック文字です。（これに対して、漢字や漢字と同じ大きさで表示される英文字、数字、記号、カタカナを**「全角文字」**と呼びます）

ご注意：ＢＡＳＩＣの命令や行番号、数値、変数名、演算子、区切り記号、ファイルスペックなどは、半角文字で入力してください。
：漢字モードでは、半角のひらがなは入力できません。半角のひらがなを使用するときは、英数・かなモードにしてください。
：漢字モードでは、グラフィックキャラクタ（⇨55ページ）は使用できません。

漢字の入力のしかた

漢字などの全角文字を入力する方法

ＢＡＳＩＣの漢字モードで全角文字（漢字など）を入力する操作は、次の順に行います。

変換画面の表示 → **読みの入力** → **変　　換** → **確　　定**（入力完了）

次の操作で変換画面を表示させます。
- **CTRL を押しながら SPACE を押す。**

（または、GRAPH を押しながら SELECT を押す。）

これでＢＡＳＩＣは、キーボードから入力された文字を漢字（全角文字）の読みとして受け取る状態になります。

ＭＳＸ turbo Ｒパソコンでは、入力された読みを「連文節変換機能」で漢字に変換します。
「連文節変換機能」とは、いくつかの「文節」（文を単語よりは大きい範囲で最も短く区切った単位）を一度に入力し、まとめて漢字に変換する機能のことです。

例）	**（ 文 ）**	**私はＭＳＸが好きです。**
	（文節）	**私は　ＭＳＸが　好きです。**
	（単語）	**私　は　ＭＳＸ　が　好き　です。**

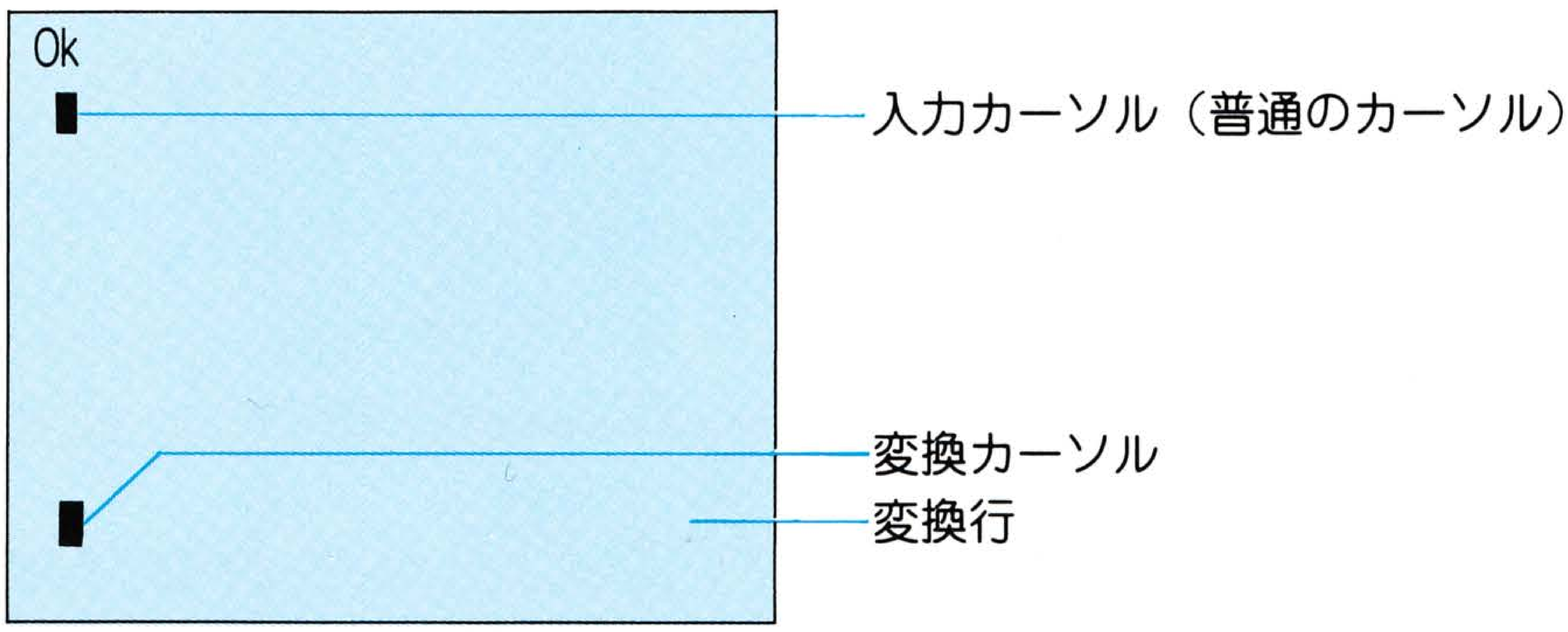
この連文節変換機能は、ＢＡＳＩＣ自身がもつ機能ではなく、ＢＡＳＩＣといっしょに動作できる「連文節変換ソフト」がもつ機能です。
以降の説明では、当社製ＭＳＸ turbo Ｒパソコンで使用している連文節変換ソフトをもとに説明しています。
この連文節変換ソフトは、「ＭＳＸ－ＪＥ」（アスキー製）に準拠しています。

※このような連文節変換ソフトのことを「日本語入力フロントエンドプロセッサ」（または日本語ＦＰ）と呼ぶこともあります。

連文節変換による文章の入力のしかた

連文節変換の画面

[CTRL] を押しながら [SPACE] を押したとき、画面の一番下の行にもうひとつカーソルが表示されます。これを**「変換カーソル」**と**「変換行」**と呼びます。

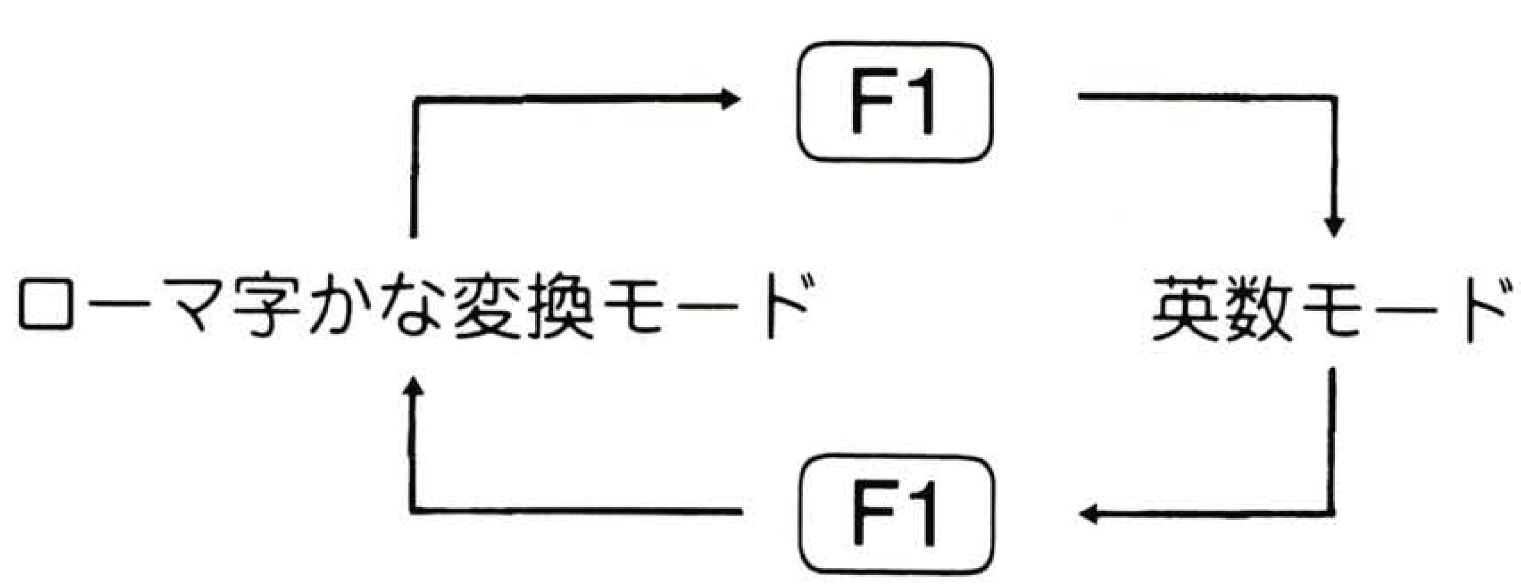


もう一度 [CTRL] を押しながら [SPACE] を押すと、変換カーソルが消え、漢字の入力ができなくなります。

連文節変換の 2 つの入力モード

連文節変換には**「ローマ字かな変換モード」**と**「英数モード」**の 2 つのモードがあります。

・変換カーソルが表示された直後は、ローマ字かな変換モードになっています。
・ローマ字かな変換モードと英数モードは、 [F1] （入力モード切替）を押すたびに切り替わります。



ローマ字かな変換モードでは、次のようにして変換行に文字を入力します。

・かなランプが消えているとき………… [B] [O] ⇨ **ぼ**

・かなランプが点灯しているとき……… [ほ] [゛] ⇨ **ぼ**

・カタカナは直接には変換行に入力できません。一度ひらがなで入力してから、 [F2] （カタカナ変換）を押してカタカナに変換します。

・数字はそのままキーを押して入力できます。

・記号のキーを押すと、かな記号が入力されます。かな記号がないキーでは、英記号が入力されます。

・英文字は原則として入力できません。英文字を入力したいときは、英数モードに切り替えてください。

英数モードで文字キーを押すと、次のように変換行に入力されます。

・かなランプが消えているときには、英文字、英記号、数字が入力されます。

・かなランプが点灯しているときには、ひらがな、かな記号が入力されます。

これ以後は、ローマ字かな変換モードで操作しているものとして説明します。

連文節変換モードでの各キーの機能

パソコンの機種によっては、連文節変換ができるワープロソフトを内蔵しているものもありますが、ＢＡＳＩＣの漢字モードでの連文節変換に使用するキーや、各キーの機能は同じではありませんから注意してください。

次ページに、ＢＡＳＩＣの漢字モードでの連文節変換で使用できる特殊キーの一覧表を掲げています。

キーの名称	機　能
F1	英数モードとローマ字かな変換モードを切り替える
F2	ひらがなをカタカナに変換する
F3	半角文字に変換する
F4	入力されたJISコードを文字に変換する
F5	すべての文節を確定する
↵	変換行の先頭の文節を確定する
∧	変換行の先頭の文節の区切りを伸ばす
∨	変換行の先頭の文節の区切りを縮める
< >	変換カーソルを左右に移動する　（変換中は未変換に戻す）
SPACE	文節変換する　（次候補文字を表示する）
ESC	入力した文字を取り消す　（変換中は未変換に戻す）
SHIFT ＋ < >	変換行の先頭または最後に変換カーソルを戻す
SHIFT ＋ SPACE	空白（スペース）の入力

※「文節」は、単語にならない範囲で最も短く文章を区切った単位

※「確定」は、変換された文字をパソコンに入力すること。

※ ひらがなを入力して F 3 を押すと、半角のカタカナに変わります。

文章の入力・変換・確定のしかた

連文節変換での変換行への文字の入力は、文章単位で行います。一度に変換行に入力できる文字数は、ひらがなで最大14文字までです。（だく音、半だく音を含む文字の場合、13文字までです。）

●入力

下のように変換行に文章を入力してください。

ぼくはぱそこんがだいすきだ。| ―変換カーソル（前景色）

このように、白地に黒で表示されている文字を**「未変換文字」**と呼びます。

◆キーを押しまちがえたときは、次のようにして訂正します。

・入力した文章をすべて取り消すときは、 ESC （**取消**）を押す。

・まちがえた文字だけを訂正するときは、左向きのカーソルキー（<）を押して変換カーソル（未変換文字の上では色で表示）をまちがえた文字に合わせ、 DEL （**削除**）を押してまちがえた文字を消し、正しい文字を入力する。

ぼくはぱそこんかだいすきだ　　変換カーソル

⬇

DEL

⬇

ぼくはぱそこんだいすきだ　　変換カーソル

⬇

G A（または か ゛）

⬇

ぼくはぱそこんがだいすきだ　　変換カーソル

・文字を挿入するときは、挿入する位置の後ろの文字に変換カーソルを合わせ、挿入したい文字を入力する。
・ BS （**後退**）を押すと、右端の文字から順に一文字ずつ消すことができる。

メモ：訂正がすんだら、▷ を押して変換カーソルを右端に戻してください。カーソル位置で文節を区切るため、そのままでは正しく変換できないことがあります。

◆読みを忘れた文字や、キーボードに無い記号などを入力するときは、ＪＩＳコード入力と呼ばれる方法を使います。ＪＩＳコード入力のしかたは、106ページ から説明しています。

●変換

未変換文字を漢字に変換するには、 SPACE （**変換**）を押します。

僕はパソコンが大好きだ

変換行に表示されていた文字が一度に漢字やカタカナに変換されます。
先頭の文節だけが、青いバックに白い字で表示され、他の部分は白地に黒で表示されます。

●確定

すべての文字が正しく変換されているときは、 F5 （**全確定**）を押し、変換行の文字を入力カーソルの位置に表示します。

◆正しく変換されていないときは、次のようにして変換し直します。

・先頭の文節からまちがっているときは、次のどれかの操作をする。

文節の区切りかたが正しいとき……正しい文字が表示されるまで SPACE （**変換**）を押す。
文節の区切りかたが短すぎるとき…正しく文節が区切られるまで**上向き**のカーソルキー（ ∧ ）を押す。
文節の区切りかたが長すぎるとき…正しく文節が区切られるまで**下向き**のカーソルキー（∨）を押す。

・先頭の文節は正しく変換されているが、その後の文字がまちがっているときは、↵ （**部分確定**）を押す。
先頭の文節だけが入力カーソルの位置に表示され、次の文節が青いバックに白で表示される。
まちがっている文字がある文節が青いバックに表示されるまで、↵ を押し続ける。あとは先頭の文節からまちがっているときと同じ操作をする。

・変換してから入力のまちがいに気付いたときは、 ESC （**取消**）または◁▷を押して変換を取り消し、まちがっている文字を訂正する。訂正がすんだら変換カーソルを右端に戻し、再び変換する。

・カタカナに変換したい文字がカタカナにならなかったときは、次のようにする。
変換カーソルをカタカナにする文節の先頭の文字に合わせ、その前までの文節を変換し、↵ を押して部分確定する。
変換カーソルをカタカナにする文節の直後の文字に合わせ、 F2 （**カタカナ変換**）を押す。

●読みを知らない漢字や記号の入力のしかた（ＪＩＳコード入力）

音読みを忘れた漢字やキーボードに無い記号を入力するときは、**「ＪＩＳコード」**と呼ばれる文字の番号を変換行に入力し、目的の文字に変換します。
次の手順で入力してください。

1 パソコンの説明書の、「ＪＩＳ第１水準漢字コード表」または「ＪＩＳ第２水準漢字コード表」で目的の文字のＪＩＳコードを調べる。（ただし、非漢字については、本書の219ページ参照）
※　ＪＩＳコードは、16進数４桁であらわします。（16進数⇨55ページ）
例）　「漢」（JIS第１水準漢字：かん）･･････････３４４１

2 [CTRL] を押しながら [SPACE] を押して、変換行を表示させる。

3 [F1] を押して、英数・かなモードに切り替える。

4 JISコードを入力する。（[3][4][4][1]）

変換行に、入力したコードが表示される。

5 [F4] （**JISコード変換**）を押す。

入力したコードにあたる文字が、変換行の左端に表示される。

6 ⏎（**部分確定**）を押して、表示された文字を確定する。

漢字入力時の注意点

●ＪＩＳ第１水準と第２水準の違いについて

常用漢字を中心に、日常の生活で一般的によく使われる文字を選んだものが「ＪＩＳ第１水準」の文字で、それ以外の文字で情報処理に必要な場合があるとして選ばれた文字が「ＪＩＳ第２水準」です。漢字コード表で目的の文字を探すときにどちらの表にある文字かが不明な場合は、まずＪＩＳ第１水準漢字コード表から探し始めてください。

●ＪＩＳ第１水準漢字コード表について

ＪＩＳ第１水準漢字コード表は、次の順番で文字に番号を付けています。

記号→英数字→ひらがな→カタカナ→ギリシャ文字→ロシア文字→けい線→漢字

漢字は、音読みを基準にアイウエオ順に、同じ読みでは画数の少ないものから順に番号が付けられています。ただし、一部の漢字（訓読みのほうが一般的に使われていると考えられたもの）については、音読みではなく訓読みを基準に順番を決められています。また、漢字によっては２通り以上の音が認められている場合もありますが、ＪＩＳコード表ではそのうちの１音だけを基準に順番が決められています。

音読みの順番で文字を探して見つからないときは、訓読み、または他の音で読んだときの順番にあたる位置も探してみてください。それでも見つからないときは、ＪＩＳ第２水準漢字コード表で探してみてください。

●ＪＩＳ第２水準漢字コード表について

ＪＩＳ第２水準漢字コード表は、次の順番で番号を付けています。

部首の画数が少ないもの（総画数　少　→　多） → **部首の画数が多いもの（総画数　少　→　多）**

部首の画数は、旧字体をもとに計算されています。したがって、新字体の画数で計算すると見つからない場合があります。（例：クサカンムリの旧字体「艸」は６画）
旧字体の部首または画数が不明な場合は、漢和辞典などでお調べください。

●全角の記号の入力

全角の記号（漢字と同じ大きさの記号）は、ＪＩＳ第１水準漢字コード表により、コード番号で入力してください。（一部の記号は、半角の記号と形が異なります。）

漢字を使える部分と使えない部分

ＢＡＳＩＣの漢字モードでは、プログラムや命令文の入力中は、どの部分にも漢字などの全角文字を入力できます。しかし、入力した部分によっては、そのプログラムや命令を実行または表示したときにエラーになることがあります。漢字などを入力するときは、そこが全角文字を入力できる部分かどうか考えながら入力してください。

全角文字を使える部分	全角文字を使えない部分
ＲＥＭ文の注釈	行番号
	命令語
文字定数	数値
	演算子
ファイル名	変数名
	区切り記号
拡張子	ファイルスペックの前後の「″」

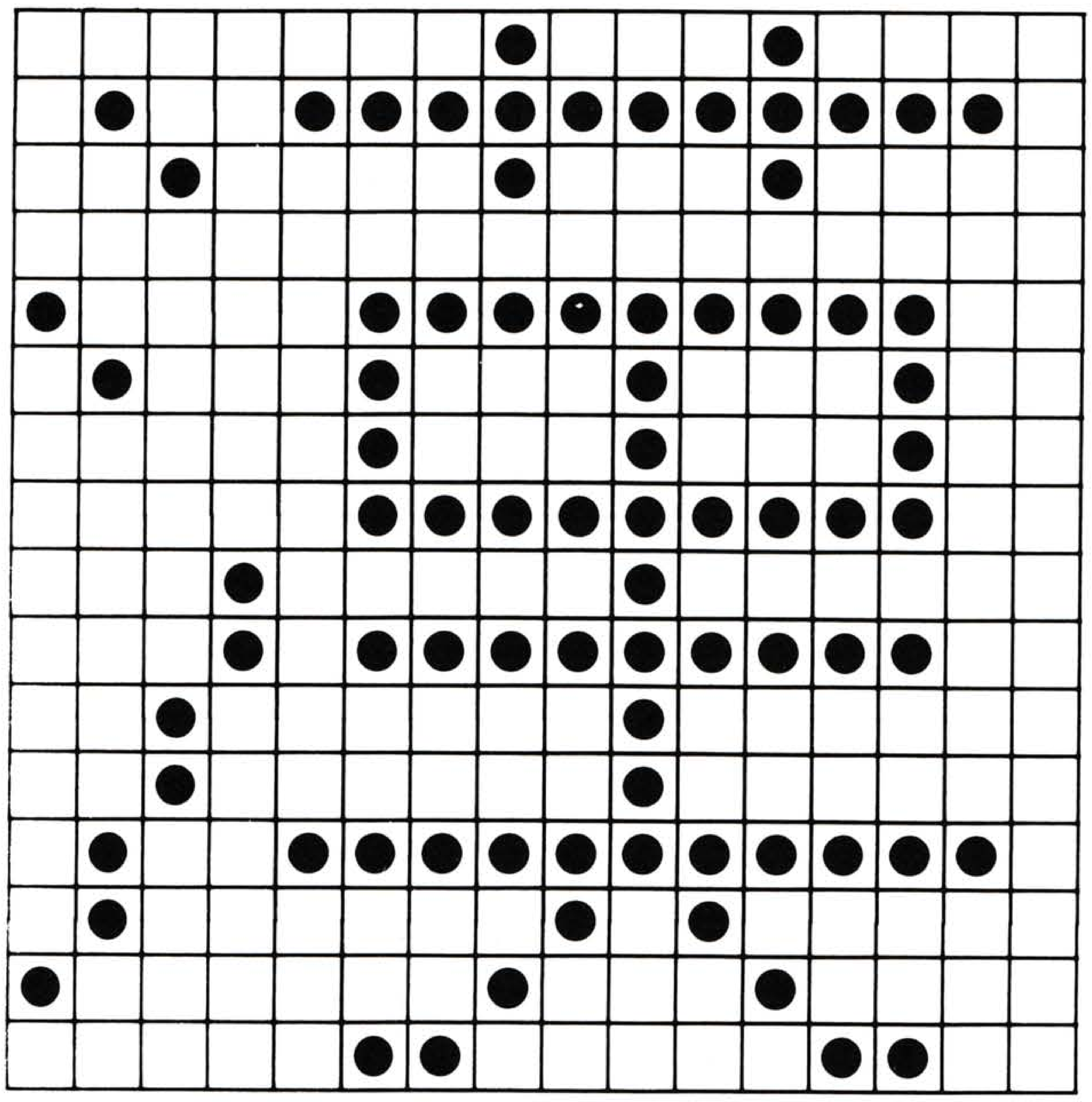
全角文字を使える部分に入力する場合も、次のことに注意してください。

- **入力できる文字数は、全角１文字を半角２文字に換算する。**
  （例：ファイル名は全角４文字まで。拡張子は全角１文字まで。）
- **英数・かなモードでは全角文字が正常に表示されない。**
- **市販ソフトなどでファイルを操作するとき、ファイル名に全角文字を使用していると正常に操作できない場合がある。**
- **漢字モードでも、全角文字の印刷には「漢字プリンタ」が必要。**

文字の形について

表示される文字の形

漢字モードの画面に表示される全角文字は、縦１６×横１６のマス目状の範囲に並んだドット（点）で表されています。（ＣＡＬＬ　ＫＡＮＪＩ０の場合。⇨111ページ）



しかし、画数が多い漢字には、１６×１６ドットでは正確に表現できないものがあります。それらの漢字は、字画の一部を省略または変更して表示しています。

表示される文字と印刷される文字の形の違い

画面に表示される文字は、パソコンが内蔵しているＲＯＭ（ロム：読み出し専用のメモリ）に記憶されている形で表されます。
これに対して、プリンタで印刷される文字は、通常、プリンタ自身が内蔵しているＲＯＭに記憶されている形で表されます。また一般にプリンタは、より正確で美しい文字を印刷するため、１文字を表すのに画面より多くのドットを使用しています。（24×24〜48×48ドット程度、機種による）
これらの理由で、画面に表示される文字とプリンタで印刷される文字は、形が大きく異なります。

特殊な文字について

人名、地名などの特殊な文字

人名や地名などの固有名詞には、ＪＩＳ漢字コード表にも含まれていない特殊な文字が使われている場合があります。
ＢＡＳＩＣには、これらの特殊な文字を表示する機能はありません。しかし、ワープロソフトや漢字プリンタには、新しい文字や記号の形を記憶して印刷する機能をもつものがあります。この機能により新しく記憶された文字を、一般に「外字」（がいじ）と呼びます。
この外字機能をもつプリンタを使えば、ＢＡＳＩＣでもＪＩＳ漢字コード表にない文字を印刷させることができる場合があります。この場合の実際の操作方法は、プリンタの機種により異なります。お使いのプリンタの取扱説明書をお読みください。

漢字モードの種類

ＢＡＳＩＣの漢字モードには、画面に表示する文字のサイズと表示のしかたによって４つの種類があります。
これらは、ＣＡＬＬ　ＫＡＮＪＩ命令を実行させるときに選びます。

・ＣＡＬＬ　ＫＡＮＪＩ０……文字サイズはヨコ16×タテ16ドット。
（または CALL KANJI）

・ＣＡＬＬ　ＫＡＮＪＩ１……文字サイズはヨコ12×タテ16（または12）ドット。

・ＣＡＬＬ　ＫＡＮＪＩ２……文字サイズはヨコ16×タテ16（または12）ドット。
インターレースモードで表示される。

・ＣＡＬＬ　ＫＡＮＪＩ３……文字サイズはヨコ12×タテ16ドット。
インターレースモードで表示される。

※「ＫＡＮＪＩ」と数字のあいだは、間隔をあけないでください。間隔をあけて入力するとエラーになります。
※インターレースモードとは、１つの画面に表示できる文字の行の数を２倍にする表示方法のことですが、一般の家庭用テレビではちらつきが大きく、実用的ではありません。

以後の説明では、ＣＡＬＬ　ＫＡＮＪＩ０およびＣＡＬＬ　ＫＡＮＪＩ１だけを使用します。

漢字モードの画面

ＢＡＳＩＣの英数・かなモードにテキスト画面とグラフィック画面があるように、漢字モードにも「漢字テキスト画面」と「漢字グラフィック画面」があります。
漢字モードでの画面の種類は、下の表のようになっています。
（この表は、漢字モード時の各画面モードの仕様をしめしています）

モード	スクリーンNo.	文字表示	図形表示	表示色	色指定
漢字テキスト	0	半角文字最大８０桁＊１ 全角文字最大４０桁	表示できない	５１２色中の１６パレット中の２色	前景色 背景色 周辺色
	1	半角文字最大８０桁＊１ 全角文字最大４０桁	スプライトのみ可能 （スプライト座標使用）＊２		
漢字グラフィック	2	半角文字最大４０桁＊３ 全角文字最大２０桁	横２５６×縦１９２ドット	５１２色中の１６パレット	水平８ドットを１単位とする領域ごとに２色
	3	半角文字最大１０桁＊３ 全角文字最大　５桁	横２５６×縦１９２ドット （６４×４８ブロック）＊４		１ブロックごとに色指定可能
	4	半角文字最大４０桁＊３ 全角文字最大２０桁	横２５６×縦１９２ドット		水平８ドットを１単位とする領域ごとに２色
	5		横２５６×縦２１２ドット		１ドットごとに色指定可能
	6	文字表示には向かない	横５１２×縦２１２ドット	５１２色中の４パレット	
	7	半角文字最大８０桁＊３ 全角文字最大４０桁		５１２色中の１６パレット	
	8	半角文字最大４０桁＊３	横２５６×縦２１２ドット	２５６色固定（スプライトは16色）	
自然画表示	10	全角文字最大２０桁		自然画表示色、１２，４９９色	＊５）
	11	文字表示には向かない		自然画表示色、１９，２６８色	
	12			自然画表示色、１９，２６８色	

＊１）ＣＡＬＬ　ＫＡＮＪＩ１、ＷＩＤＴＨ　８０実行時の数値。

＊２）スプライト座標は、Ｘは－３２～２５５、Ｙは－３２～１９１（スクリーン５以上では、－３２～２１１）。

＊３）ＣＡＬＬ　ＫＡＮＪＩ１実行時の数値。プログラムモードでのみ表示可能。

＊４）１ブロックは４×４ドットの領域。

＊５）スクリーン１１・１２では、図形表示に２５６色分の色コードを指定できますが、色表示の方法が他の画面と異なるため、同じコードによる色指定をしても他の画面とは異なる色になったり、まったく表示されなかったりします。図形表示はスクリーン１０までの画面モードでしてください。

漢字モードの文字表示位置座標

漢字モードでは、漢字テキスト画面、漢字グラフィック画面のどちらでも、ＬＯＣＡＴＥ文やＰＲＩＮＴ文で画面の指定した位置に文字を表示することができます。

このとき、画面の表示位置は、半角文字一文字を単位とする桁と行で指定します。

（**例**：ＫＡＮＪＩ０でヨコ座標の指定を5にしたときの表示位置）

```
１０ＬＯＣＡＴＥ　５、２↵
２０ＰＲＩＮＴ"田中君！"↵
```



漢字モードでは、漢字テキスト画面、漢字グラフィック画面のどちらでも、文字の表示位置を上のように半角文字単位で指定します。

漢字で住所氏名を表示しよう

漢字テキストモードを使ったプログラムです。
ＩＮＰＵＴ文で住所氏名を漢字で入力し、色を変えて画面に表示します。



※プログラムを止めるときは、CTRL を押しながら STOP を押します。

※プログラムの漢字以外の部分は、半角文字で入力してください。全角文字で入力するとエラーになります。

リストと解説

プログラム	解説
1 'save"ｼﾞｭｳｼｮ.bas"	
100 CALL KANJI	漢字モード０に切り替える
110 COLOR 15,4	表示色の初期設定
120 CALL CLS	漢字モードでの画面消去
130 PRINT:COLOR 15	前景色の再設定（白）
140 INPUT "氏名";NM$	氏名住所の入力
150 INPUT "住所";J$	（140〜150行）
160 PRINT:COLOR 7	改行：前景色の変更（淡い青）
170 PRINT NM$	氏名住所の表示
180 PRINT J$	（170〜180行）
190 GOTO 130	

参　考

・漢字プリンタを接続している場合、160行〜180行のＰＲＩＮＴ文をＬＰＲＩＮＴ文に変えることにより、入力した氏名・住所を印刷することができます。
・131ページ にかんたんな住所録作成のサンプルプログラムを掲載しています。参考にしてください。

新しい命令の解説

漢字モードの文字列

漢字モードでは、文字列に全角文字を使うことができます。
（100,140,150,170,180行）

英数・かなモードでは、１つの文字列に 255文字までの半角文字を使うことができました。
漢字モードでは、１つの文字列に全角文字を 127文字まで使うことができます。
（全角文字１文字は、半角文字２文字にあたります）

ただし、次の場合には文字列に全角文字を使うことができませんから注意してください。

例）
```
OPEN "GRP:" AS #1:PRINT#1,"文字列"
```

CALL CLS（コール・シーエルエス）

漢字モードの漢字テキスト画面では、英数・かなモードで使えたＣＬＳ命令が使えません。
（使うとエラーになります）
漢字テキスト画面を消去するときは、ＣＡＬＬ　ＣＬＳ命令を使います。
ＣＡＬＬ　ＣＬＳは、漢字グラフィック画面でも、英数・かなモードでも使えます。（120行）
※漢字グラフィック画面では、ＣＬＳ命令も使えます。

漢字モードの文字表示色

英数・かなモードでは、前景色の指定を変えると画面に表示されているすべての文字の色が変わります。同時に画面に表示する文字の色が１色だけに限られているからです。
漢字モードでは、１文字単位で表示色を指定できます。
前景色の指定を変えても、先に表示した文字の色は変わりません。（130行、160行）

漢字グラフィック画面で文字表示

漢字グラフィック画面で、図形に漢字を重ねて表示するプログラムです。



※プログラムを止めるときは、CTRL を押しながら STOP を押します。

リストと解説

```
1 'save"カンジ GRP.bas"
100 CALL KANJI:SCREEN 5
110 CALL PALETTE:COLOR ,4
120 FOR R=5 TO 100 STEP 5
130   CIRCLE (125,103),R,8
140 NEXT R
150 LOCATE 9,6:PRINT "漢字B
ASIC"
160 GOTO 160
```

- 100：スクリーン5の画面に切り替える
- 120〜140：5から 100まで5ずつ半径を増しながら、赤で同心円を描く
- 150：同心円に重ねて文字を表示する

新しい命令の解説

漢字グラフィック画面の文字表示

漢字グラフィック画面に文字を表示するときには、漢字テキスト画面と同じく、次の２つの命令を使用します。

LOCATE…………	表示位置の指定
PRINT……………	文字列の表示

・表示位置の指定に使用する位置座標も、漢字テキスト画面と同じく、半角文字単位の桁と行で指定します。

・漢字グラフィック画面では、WIDTH文は使用できません。
１行に表示できる文字数（桁数）は、最大で１行20文字を限度として、漢字モードの種類による文字のドット数と画面モードによる画面のヨコ方向のドット数で決定されます。

例）

漢字モード０（ヨコ16ドット）で画面モードがスクリーン５（ヨコ 256ドット）のとき
256÷16＝16　で　１行の表示文字数は、全角文字16桁。（半角文字32桁）

漢字モード１（ヨコ12ドット）で画面モードがスクリーン７（ヨコ 512ドット）のとき
512÷12＝42 だが１行の表示文字数は、全角文字40桁。（半角文字80桁）

おまけのプログラム

```
1 'save"ﾃﾝﾁ.bas"
100 CALL KANJI:SCREEN 5
110 COLOR 4,0,0:CALL CLS
120 LOCATE 12,5
130 PRINT "天地無用"
140 FOR N=0 TO 79
150   X=96+N
160   FOR M=0 TO 15
170     COPY (X,80+M)-(X,80+M
) TO (X,112-M)
180   NEXT M
190 NEXT N
200 GOTO 200
```

漢字グラフィック画面を使ったおまけのプログラムです。
特に解説はしませんが、実行には少し時間がかかります。

※プログラムを止めるときは、CTRL を押しながら STOP を押してください。

※COPY（コピー）命令のこのような使いかたについては、この本では説明していません。興味がある人は、市販のMSX　BASICの解説書で調べてみてください。

「入門」のおわりに

さて、ＢＡＳＩＣについての説明は、これで一応おしまいです。お疲れさまでした。
どうですか？　思ったよりカンタンだったでしょう？

この本では、ＢＡＳＩＣのもつすべての命令について説明することよりも、むしろＢＡＳＩＣに親しんでもらうことに重点をおきました。
これまでに説明してきたことは、ＢＡＳＩＣがもつたくさんの命令や関数のほんの一部分に過ぎません。
言い換えれば、あなたはいまＢＡＳＩＣへの「入門」を終わったばかりなのです。

「もっとＢＡＳＩＣの勉強をしたい」という人は、この本の 175ページから始まる「ＢＡＳＩＣリファレンス表」や、書店やパソコンの専門店で販売されているＭＳＸ　ＢＡＳＩＣの解説書などを使って、どうぞ勉強を続けてください。
また、「この辺で一息入れたいな」という人は、次のページから始まる「第６章　サンプルプログラム集」の中から、好きなプログラムを選んで試してみてください。

ＢＡＳＩＣをマスターするひとつのコツは、あせらず、あきらめずに根気よく続けるということです。「私はＢＡＳＩＣをマスターした」と言える日をめざして、どうかあきらめずに頑張ってください。

第6章

楽しむためのＢＡＳＩＣだという人のためのサンプル集です。

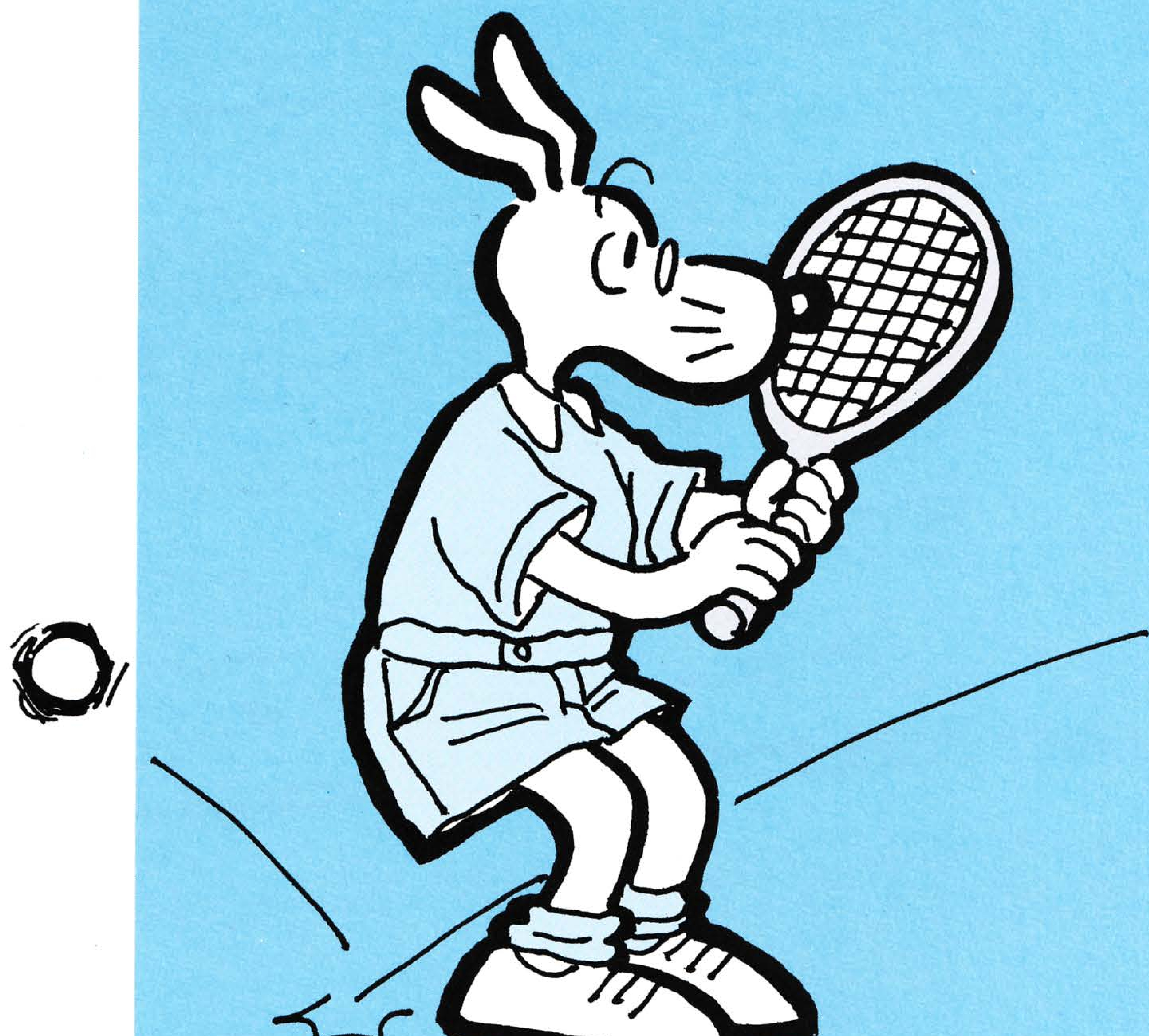
サンプルプログラム集

ＵＦＯ …………………………………………………… 120

ヘリコプター ………………………………………… 123

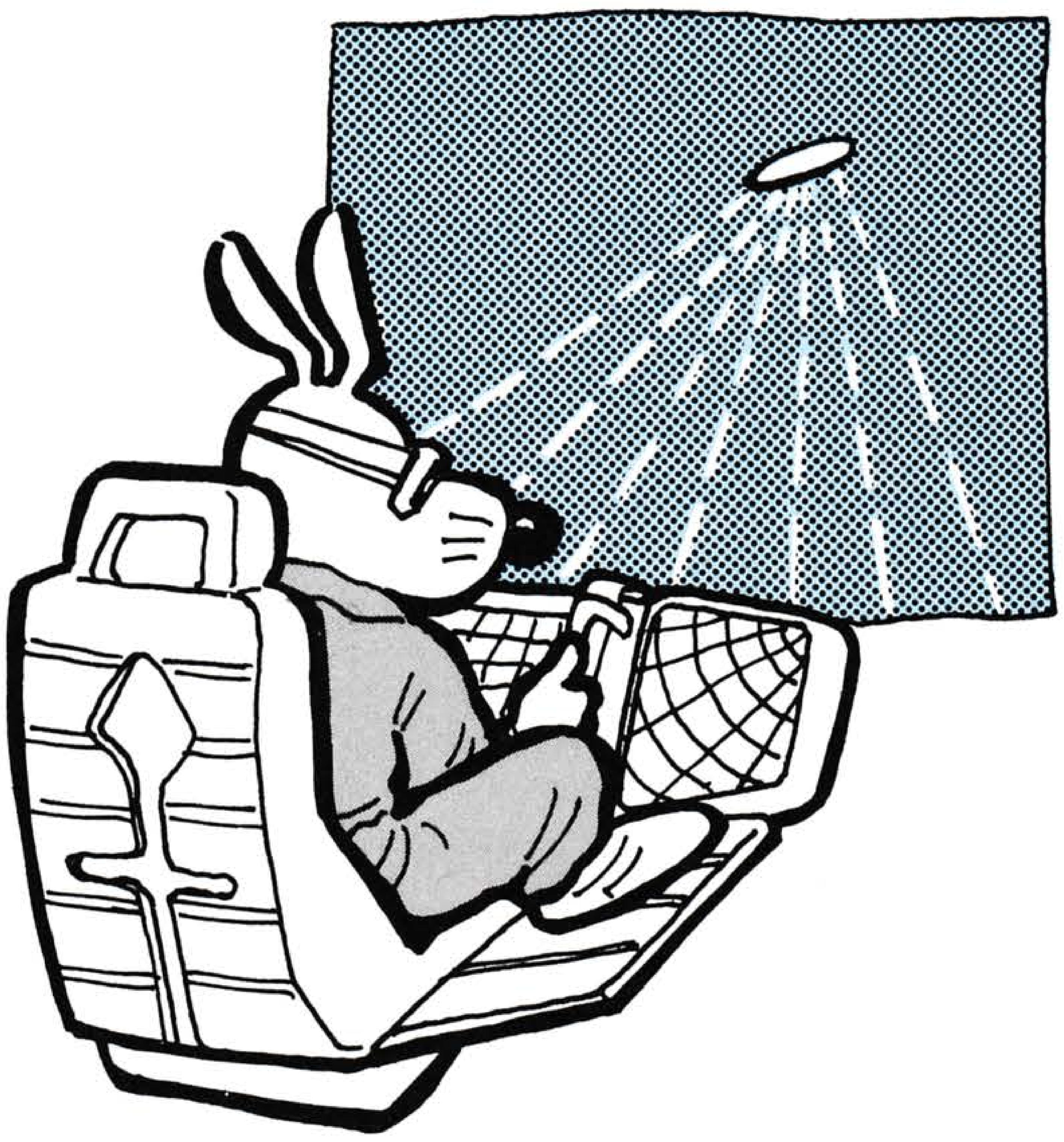
お絵かきプログラム ………………………………… 127

住所録作成プログラム……………………………… 131



UFO

上空に飛んでくるＵＦＯを、砲台を左右に動かしながらミサイルで撃ち落とすゲームです。うまくＵＦＯを撃ち落とすと得点になりますが、はずしてしまうと得点がマイナスされます。１機のＵＦＯに対して１発しかミサイルを撃てませんから、慎重に狙ってください。



砲台の移動は左右のカーソルキーを、ミサイルの発射は SPACE を押します。

プログラムを途中で止めるときは、CTRL を押しながら STOP を押したあと、ダイレクトモードで次の命令を実行させてください。

SCREEN 5
（スプライトパターンの消去）

```
1 'save"UFO.BAS"
100 CALL ANK
110 SCREEN 1,1:WIDTH 29:COLOR
:COLOR 15,4,7:CLS:KEY OFF
120 FOR K=0 TO 2
130   B$=""
140   FOR J=0 TO 7
150     READ A$
160     B$=B$+CHR$(VAL("&B"+A
$))
170   NEXT J
180   SPRITE$(K)=B$
190 NEXT K
200 LOCATE 3,15:PRINT"とくてん"
210 ON SPRITE GOSUB 470
220 SPRITE ON
230 YN=-18:YY=209
240 X=150:T=1
250 DX=2
```

- 100：ＢＡＳＩＣを英数・かなモードに
- 110：画面を初期設定する
- 120〜190：スプライトパターンを定義する
- 210：スプライト同志が衝突したら 470行へ
- 220：以後、スプライトの衝突をチェックする
- 230〜250：各種の変数の値を初期設定する

```
260 PUT SPRITE 0,(XX,YY),W,0
270 IF W=4 THEN W=10 ELSE W=4

280 XX=XX+DX:IF XX<239 THEN 3
10
290 XX=0:DX=RND(1)*3+1:YY=RND
(1)*100:T=INT(T-1):LOCATE 6,1
6:PRINT T:FG=1
300 N=N+1:IF N>50 THEN 410
310 ST=STICK(0):IF ST=0 THEN
GOTO 360
320 IF ST=3   THEN X=X+4
330 IF X >239 THEN X=239
340 IF ST=7   THEN X=X-4
350 IF X <0   THEN X=0
360 PUT SPRITE 1,(X,174),8,1
370 PUT SPRITE 2,(X2,Y2),10,2

380 Y2=Y2-6:IF Y2>-18 THEN 26
0
390 Y2=-18:IF FG*STRIG(0)=0 T
HEN 260
400 FG=0:Y2=157:X2=X+2:GOTO 2
60
410 '* ｹﾞｰﾑ ｵｰﾊﾞｰ *
420 SCREEN 2:SCREEN 1
430 LOCATE 3,7:PRINT"ｹﾞｰﾑ ｵｰﾊ
ﾞｰ !"
440 LOCATE 5,10:PRINT"とくてん=";
T
450 PLAY"O4V8L4DCFG8O5C8O4AG8
O5C8O4AFGDC"
460 KEY ON:PRINT:END
```

- 260〜290: ＵＦＯの移動と色の変化の処理
- 300: ゲームが終了したかどうかの判定
- 310〜360: カーソルキーと SPACE が押されているかどうかを調べ、カーソルキーが押されていたら砲台を左右に動かし、SPACE が押されていたら280行へ
- 370〜400: ミサイルの発射と移動の処理
- 410〜460: ゲーム終了時の処理

第6章 サンプルプログラム集

```
470 '* ﾒｲﾁｭｳ *
480 IF YY=209 THEN RETURN
490 T=T+DX*2:PLAY"O3S9M2000C
"
500 YY=209:Y2=-18
510 RETURN
520 '* UFO ﾊﾟﾀｰﾝ *
530 DATA 00011000
540 DATA 00111100
550 DATA 01111110
560 DATA 11111111
570 DATA 00100100
580 DATA 00100100
590 DATA 01000010
600 DATA 10000001
610 '* ﾎｰﾀﾞｲ ﾊﾟﾀｰﾝ *
620 DATA 00111100
630 DATA 00111100
640 DATA 00111100
650 DATA 00111100
660 DATA 00111100
670 DATA 00111100
680 DATA 11111111
690 DATA 11111111
700 '* ﾐｻｲﾙ ﾊﾟﾀｰﾝ *
710 DATA 00000000
720 DATA 00000000
730 DATA 00010000
740 DATA 00010000
750 DATA 00010000
760 DATA 00010000
770 DATA 00000000
780 DATA 00000000
```

ミサイルがＵＦＯに当たったときの処理（470〜510）

ＵＦＯと砲台、ミサイルのスプライトパターン（520〜780）

ヘリコプター

あなたはヘリコプターの操縦を習っている訓練生です。いじわるな教官から、燃料切れ寸前のヘリコプターで移動するプラットフォームの上に着陸するよう命令されました。みごとに着陸してみせて、教官に「マイッタ！」と言わせましょう。



※ゲームが始まったとき、ヘリコプターはエンストをおこしています。[∧]ですぐに上昇してください。

※なにしろオンボロのヘリコプターですから、そっと着陸してください。いきおいがついたままプラットフォームに着陸すると、バラバラになってしまいます。

※着陸に成功すると、残っている燃料の分だけ得点になります。

```
1 'save"ﾍﾘｺﾌﾟﾀｰ.BAS"
1000 CALL ANK
1010 SCREEN 5,1:COLOR:COLOR 1
5,4,7:CLS
1020 FOR T=0 TO 2
1030   B$=""
1040   FOR N=0 TO 7
1050     READ A$
1060     B$=B$+CHR$(VAL("&b"+
A$))
1070   NEXT N
1080   SPRITE$(T)=B$
1090 NEXT T
1100 OPEN "grp:" AS #1
1110 ON SPRITE GOSUB 1430
1120 TK=0
```

- 1000：BASICを英数・かなモードに
- 1010：スクリーン５の画面でスプライトの種類を８×８ドット拡大にし、画面を初期設定する
- 1020〜1090：スプライトパターンを３個定義する
- 1100：グラフィック画面を文字表示可能に
- 1110：ヘリコプターがプラットフォームに衝突したら1430行へ
- 1120：得点をあらわす変数ＴＫの値を０に

```
1130 '--- ｹﾞｰﾑ ｽﾀｰﾄ
1135 FOR Q=1 TO 6000:NEXT
1140 X=0:Y=0:U=1:F=1:P=100
1150 X1=-16:Y1=196
1160 PRESET ( 84,8):PRINT #1,
"ﾄｸﾃﾝ" ;TK
1170 PRESET (176,8):PRINT #1,
"ﾈﾝﾘｮｳ";P
1180 SPRITE ON
1190 '--- ﾍﾘｺﾌﾟﾀｰｦ ｳｺﾞｶｽ
1200 Y=Y-U/5:X=X+F
1210 IF X<0   THEN X=  0:F=0
1220 IF X>240 THEN X=240:F=0
1230 IF Y<0   THEN Y=  0:U=0
1240 IF Y>196 THEN Y=196
1250 PUT SPRITE 1,(X,Y),10,1
1260 IF Y=196 THEN GOSUB 1430

1270 PLAY"T255V15O4S8M1000B64
"
1280 '--- ﾌﾟﾗｯﾄﾌｫｰﾑｦ ｳｺﾞｶｽ
1290 X1=X1+2:IF X1=256 THEN X
1=-14
1300 PUT SPRITE 2,(X1,Y1),13,
2
1310 '--- ｶｰｿﾙｷｰｦ ｼﾗﾍﾞﾙ
1320 K=STICK(0)
1330 IF K=0 THEN U=U-1:GOTO 1
410
1340 P=P-1:IF P<=0 THEN P=0:U
=U-1:F=0
1350 PRESET (216,8):PRINT #1,
P
1360 IF P=0 THEN 1420
1370 IF 7<K OR  K<3 THEN U=U+
1
```

各変数に設定する初期値

得点と残りの燃料を画面に表示する

以後、スプライトの衝突による割り込み処理を行う

表示する位置を決める

ヘリコプターを表示する

地面にぶつかっていたら、1430行へ

飛行中の音を鳴らす

表示する位置を決める

プラットフォームを表示する

押されているカーソルキーを調べ、各方向への速度と燃料の残量を変える

燃料の残量を表示する

燃料が残っていないなら、1420行へ

```
1380 IF 1<K AND K<5 THEN F=F+
1
1390 IF 3<K AND K<7 THEN U=U-
1
1400 IF 5<K         THEN F=F-
1
1410 '
1420 GOTO 1190
1430 '--- しょうとつ
1440 SPRITE OFF
1450 IF X<X1-4 OR X1+4<X THEN
 1500
1460 IF 184<Y              THEN
 1500
1470 '--- ちゃくりく せいこう
1480 PUT SPRITE 1,(X,182),10,
1
1490 TK=TK+P:Z$="ﾁｬｸﾘｸ ｾｲｺｳ!"
:GOTO 1550
1500 '--- ちゃくりく しっぱい
1510 PLAY"T120V15O3S9M15000C1
"
1520 TK=TK-100+P
1530 Z$="ﾁｬｸﾘｸ ｼｯﾊﾟｲ!"
1540 PUT SPRITE 0,(X,Y+4),9,0

1550 '--- 1ﾗｳﾝﾄﾞ しゅうりょう
1560 PRESET(75,84):PRINT #1,Z
$
1570 IF 500<TK THEN Z$="  ﾏｲﾘ
ﾏｼﾀ!!  ":GOTO 1650
1580 IF TK<-200 THEN Z$="ｷﾐ は
ﾗｸﾀﾞｲした!":GOTO 1650
1590 FOR T=0 TO 6000:NEXT
```

- 1420: 1190行に戻る
- 1440: スプライトの衝突による割り込みを禁止する
- 1450: プラットフォームの中心から4ドット以上外れていたら1500行へ
- 1460: 勢いがつきすぎていたら1500行へ
- 1480: プラットフォーム上にヘリコプターを表示する
- 1490: 成功したときの得点と表示を設定する
- 1510: 爆発音を鳴らす
- 1520–1530: 失敗したときの得点と表示を設定する
- 1540: 爆発のスプライトを表示する
- 1560: 結果を表示する
- 1570: 得点がプラス500を越えたときは、別の表示を設定し1650行へ
- 1580: 得点がマイナス200を越えたときは、別の表示を設定し1650行へ
- 1590: 表示を読むための時間をあける

```
1600 PUT SPRITE 0,(0,217),,0
1610 PUT SPRITE 1,(0,217),,1
1620 PUT SPRITE 2,(0,217),,2
1630 CLS
1640 RETURN 1130
1650 '--- ｹﾞｰﾑ しゅうりょう
1660 PRESET(80,102):PRINT#1,Z
$
1670 PRESET ( 84,8):PRINT #1,
"ﾄｸﾃﾝ" ;TK
1680 FOR I=0 TO 9999:NEXT
1690 END
1700 '--- ｽﾌﾟﾗｲﾄ ﾃﾞｰﾀ
1710 DATA 00010000
1720 DATA 01011010
1730 DATA 01111110
1740 DATA 11111111
1750 DATA 00111111
1760 DATA 01011010
1770 DATA 11001001
1780 DATA 00001000
1790 '
1800 DATA 01111111
1810 DATA 00001100
1820 DATA 11010010
1830 DATA 11011111
1840 DATA 01111111
1850 DATA 00111110
1860 DATA 00100100
1870 DATA 01111110
1880 '
1890 DATA 00000000
1900 DATA 11111111
1910 DATA 11000011
1920 DATA 10100101
1930 DATA 10011001
1940 DATA 10100101
1950 DATA 11000011
1960 DATA 11111111
```

すべてのスプライトを消し、1130行へ戻ってゲームを再開する（1600〜1640）

最終結果と最終得点を表示し、少し時間をあけてからプログラムの実行を終了する（1660〜1690）

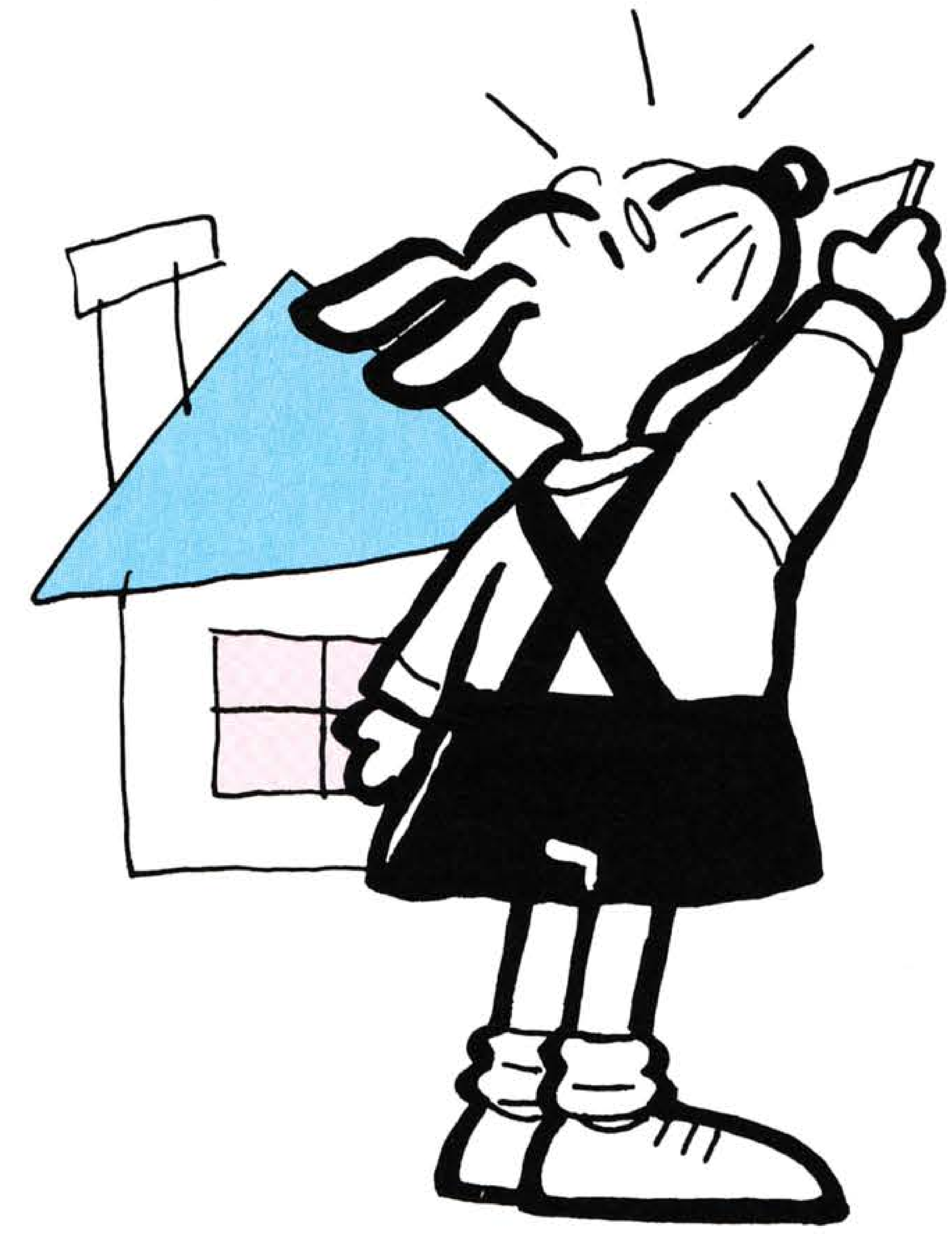
爆発のスプライトパターン（1710）

ヘリコプターのスプライトパターン（1800）

プラットフォームのスプライトパターン（1890）

お絵かきプログラム

カーソルキーを使って画面に絵を描く「お絵かきプログラム」です。
ごく限られた機能しかありませんが、みなさんの手で改良してみてください。



※プログラムを止めるときは、CTRL を押しながら、STOP を押してください。

キーの機能

・色指定（透明と黒を含む16色指定可能）‥‥数字キーおよびA～Fの文字キー
・カーソルの移動（8方向）‥‥‥‥‥‥‥‥カーソルキー
・直線を描くモードと描かないモードの切替‥ SPACE
・同じ色の線で囲まれた部分に色を塗る‥‥‥ P
・画面を初期化する‥‥‥‥‥‥‥‥‥‥‥‥ DEL

使いかた

画面下部の色見本を参考に色を指定し、画面に表示されるカーソルを動かすことにより、図形を描きます。
カーソルの色と形が、現在のモードと色指定をあらわします。

・×形のカーソル‥‥‥‥‥‥‥‥‥‥‥‥線を描かないモード ┐
・十形のカーソル‥‥‥‥‥‥‥‥‥‥‥‥線を描くモード ┘ SPACE で切り替えます。
・カーソルの色‥‥‥‥‥‥‥‥‥‥‥‥‥現在の指定色
・点線のカーソル‥‥‥‥‥‥‥‥‥‥‥‥透明または黒の色指定（線の消去に使用）

ある色で完全に囲んだ部分の中にカーソルを入れ、同じ色を指定してから P を押すと、その部分を塗りつぶします。
囲んだ線と異なる色を指定して P を押したり、完全に囲んでいない部分で P を押すと、画面全体を塗りつぶしてしまいます。あやまって画面を塗りつぶしてしまったときは、 DEL を押して画面を初期化し、初めから描き直してください。

```
1 'save"ｵｴｶｷ.BAS"
100 CALL ANK:SCREEN 5,0
110 OPEN "grp:" AS #1
120 '--- ｽﾌﾟﾗｲﾄ ﾉ ﾃｲｷﾞ
130 FOR I=0 TO 255:SPRITE$(I)
="":NEXT
140 FOR I=0 TO 31:PUT SPRITE
I,(0,217),,I:NEXT
150 FOR I=0 TO 3
160   B$=""
170   FOR N=0 TO 7
180     READ A$
190     B$=B$+CHR$(VAL("&B"+A
$))
200   NEXT N
210   SPRITE$(I)=B$
220 NEXT I
230 '--- ｶﾞﾒﾝｦ ｴｶﾞｸ
240 COLOR 15,0,0:CLS
250 YM=196
260 FOR C=0 TO 15
270   XM=C*16+4
280   PRESET (XM+1,YM):PRINT
#1,HEX$(C)
290   LINE(XM,YM+8)-STEP(7,7)
,C,BF
300 NEXT
310 LINE (0,0)-(255,195),15,B
320 X=128:Y=98:D=0:B=0:C=15:C
S=C
330 '--- ｵｴｶｷ ﾙｰﾁﾝ
340 CS=C:IF CS=<1 THEN CS=15
350 PUT SPRITE 0,(X,Y),CS,B+D

360 K$=INPUT$(1)
```

- 100: 英数・かなモード、スクリーン5を指定
- 110: 表示された画面を文字表示可能にする
- 130–140: 各スプライトパターンとスプライト面を初期化する
- 150–220: 4種類のカーソルのスプライトパターンを定義する
- 240–310: 初期画面を表示する
- 320: 各変数の値を初期設定する
- 350: カーソルのスプライトを表示する
- 360: キー入力を読む

```
370 IF K$=CHR$(&H7F) THEN 240

380 IF K$=CHR$(&H20) THEN D=1
-D:GOTO 330
390 IF "0"=<K$ AND K$=<"1" TH
EN C=VAL(K$)      :B=2:GOTO 33
0
400 IF "2"=<K$ AND K$=<"9" TH
EN C=VAL(K$)      :B=0:GOTO 33
0
410 IF "A"=<K$ AND K$=<"F" TH
EN C=VAL("&h"+K$):B=0:GOTO 33
0
420 IF "a"=<K$ AND K$=<"f" TH
EN C=VAL("&h"+K$):B=0:GOTO 33
0
430 IF K$="P" OR K$="p" THEN
GOSUB 520:GOTO 330
440 IF K$=CHR$(&H1C) AND     X
<250 THEN X=X+1:GOTO 490
450 IF K$=CHR$(&H1D) AND -3<X
     THEN X=X-1:GOTO 490
460 IF K$=CHR$(&H1E) AND -4<Y
     THEN Y=Y-1:GOTO 490
470 IF K$=CHR$(&H1F) AND     Y
<189 THEN Y=Y+1:GOTO 490
480 GOTO 330
490 '
500 IF D=1 THEN PSET (X+4,Y+5
),C
510 GOTO 330
520 '----- ﾍﾟｲﾝﾄ ｻﾌﾞﾙｰﾁﾝ
530 PRESET (X+4,Y+5):PAINT (X
+4,Y+5),C
540 RETURN
```

- 370: DEL が押されていたときの処理
- 380: SPACE が押されていたときの処理
- 390: 0 か 1 が押されていたときの処理
- 400: 2 ～ 9 が押されていたときの処理
- 410～420: A ～ F または a ～ f が押されていたときの処理
- 430: P が押されていたときの処理
- 440～470: カーソルキーが押されていたときの処理
- 480: 330行へ戻る
- 500: 線を描く処理
  描画モードなら指定色で点を描く
- 510: 330行へ戻る
- 530～540: 塗りつぶしの処理
  指定色で色を塗り、元の処理に戻る

```
550 '--- ｽﾌﾟﾗｲﾄ ﾊﾟﾀｰﾝ *
560 DATA 00000000
570 DATA 01000001
580 DATA 00100010
590 DATA 00010100
600 DATA 00001000
610 DATA 00010100
620 DATA 00100010
630 DATA 01000001
640 '
650 DATA 00000000
660 DATA 00001000
670 DATA 00001000
680 DATA 00001000
690 DATA 01111111
700 DATA 00001000
710 DATA 00001000
720 DATA 00001000
730 '
740 DATA 00000000
750 DATA 01000001
760 DATA 00000000
770 DATA 00010100
780 DATA 00000000
790 DATA 00010100
800 DATA 00000000
810 DATA 01000001
820 '
830 DATA 00000000
840 DATA 00001000
850 DATA 00000000
860 DATA 00001000
870 DATA 01010101
880 DATA 00001000
890 DATA 00000000
900 DATA 00001000
```

カーソルのスプライトパターンのデータ

住所録作成プログラム

ＢＡＳＩＣの漢字モードを使った、住所録作成プログラムです。これまでに説明していないフロッピーディスクのファイルを操作する命令を使いますが、かんたんな説明をつけています。



実行させせると画面にメニューが表示されますから、番号を入力して作業を選んでください。

※このプログラムを実行させるときは、あらかじめＡドライブにフォーマット済みで書き込み可能なフロッピーディスクを入れておいてください。

```
1 'save"ｼﾞｭｳｼｮﾛｸ.BAS"'
100 '*** ｼﾞｭｳｼｮﾛｸ ﾉ ｻｸｾｲ ***
110 '* ｻｷﾞｮｳﾊﾞﾝｺﾞｳ ﾉ ｼﾃｲ *            ― 作業メニューの処理
120 CALL KANJI1:CLEAR 800           ― 漢字モード１でメモリを初期化する
130 MAXFILES=2                      ― 同時に使用するファイル数の上限を宣言する
140 CALL PALETTE:COLOR 15,4,
7:WIDTH 36:KEY OFF                  ― 画面の初期設定
150 LOCATE 6,1:PRINT "＊＊＊
  住所録　＊＊＊"                    ― 作業のメニューを表示する
160 LOCATE 8,3:PRINT "１――
―新規作成"
170 LOCATE 8,4:PRINT "２――
―追加"
180 LOCATE 8,5:PRINT "３――
―全データ表示"
190 LOCATE 8,6:PRINT "４――
―検索・表示"
200 LOCATE 8,7:PRINT "５――
―削除"
```

第６章 サンプルプログラム集

```
210 LOCATE 8,8:PRINT "6――
―終了"
220 LOCATE 6,10:INPUT "作業
番号の入力（1―6）";N1$
230 CALL KACNV(N$,N1$):N=VAL
(N$)
240 IF N<1 OR N>6 THEN 140
250 ON N GOTO 270,360,600,59
0,580,860
260 '* ｼﾝｷｻｸｾｲ *
270 OPEN "JYUUSHO.FAT" AS #1
280 FIELD #1,255 AS FT$
290 FF$=""
300 FOR F=1 TO 255
310 FF$=FF$+"0"
320 NEXT F
330 LSET FT$=FF$:PUT #1,1
340 CLOSE #1
350 '* ﾃﾞｰﾀ ﾉ ﾄｳﾛｸ *
360 OPEN "JYUUSHO.FAT" AS #1
370 FIELD #1,255 AS FT$
380 GET #1,1
390 OPEN "JYUUSHO.DAT" AS #2
 LEN=84
400 FIELD #2,16 AS HN$,28 AS
 HT$,40 AS HJ$
410 FOR C1=1 TO 255
420 F1$=MID$(FT$,C1,1):IF F1
$="1" THEN 530
430 MID$(FT$,C1,1)="1"
440 CALL CLS:NM$=""
450 LOCATE 2,2:INPUT "氏名";
NM$
460 IF NM$="" THEN 560
470 LOCATE 2,3:INPUT "Tel ";
TL$
```

- 210: （画面表示）
- 220: 作業番号の入力を求める
- 230: 入力された数字を半角文字にし、数値に変換する
- 250: 数値（作業番号）によってジャンプする
- 270: ディスクに索引用のファイルを作り、ファイル番号1とする
- 280: ファイル1の変数領域の 255バイトを「ＦＴ$」とする
- 290–320: 変数ＦＦ$の値を 255個の「0」にする
- 330: 変数領域ＦＴ$にＦＦ$の値を入れ、変数領域の値をファイルに書き込む
- 340: ファイルを閉じる
- 360: ファイル1として索引用のファイルを開く
- 370: ファイル1の変数領域の255バイトをＦＴ$に
- 380: 変数領域にファイル1のデータを読み込む
- 390: 住所録用のファイルを作り、ファイル番号2とする
- 400: ファイル2の変数領域を3つに分ける
- 420–430: 索引（ＦＴ$）の中で空きマーク（0）を探し、最初に見付けた空きマークを「1」に書き換える
- 440–450: 登録するデータの入力を求める
- 460: 「氏名」の入力で ⏎ だけが押されたら、 560行にジャンプする
- 470: 各項目のデータの入力を求める

```
480 LOCATE 2,4:INPUT "住所";
J$
490 LOCATE 2,6:INPUT "登録 1
:Ok/0:NG";A$
500 IF A$="１" OR A$="1" THE
N 510 ELSE 450
510 LSET HN$=NM$:LSET HT$=TL
$:LSET HJ$=J$
520 PUT #1,1:PUT #2,C1
530 NEXT C1
540 PRINT "   DATA AREA OVER
FLOW  !! "
550 FOR C2=0 TO 1500:NEXT C2
560 CLOSE:GOTO 140
570 '* ﾋｮｳｼﾞ･ｹﾝｻｸ･ｻｸｼﾞｮ *
580 LOCATE 6,12:INPUT "削除
データ（氏名）";D1$:GOTO 600
590 LOCATE 6,12:INPUT "表示
データ（氏名）";D1$
600 CALL CLS
610 OPEN "JYUUSHO.FAT" AS #1
620 FIELD #1,255 AS FT$
630 GET #1,1
640 OPEN "JYUUSHO.DAT" AS #2
 LEN=84
650 FIELD #2,16 AS HN$,28 AS
 HT$,40 AS HJ$
660 FOR C1=1 TO 255
670 F1$=MID$(FT$,C1,1)
680 IF F1$="0" THEN 800
690 GET #2,C1
700 IF N=3 THEN 730
710 CALL KINSTR(A,HN$,D1$)
720 IF A=0 THEN 800
730 PRINT HN$:PRINT HT$:PRIN
T HJ$:PRINT
```

490–500: このままファイルに登録してよいかどうか確かめ、ダメなら450行に戻る

510: 入力されたデータをファイル２の変数領域に転送する

520–530: ２つの変数領域の値をそれぞれのファイルに書き込み、次のデータの入力を求める

540: データを入力しすぎたときの表示

560: すべてのファイルを閉じ、メニューに戻る

580: 削除のときは氏名の入力を求め、600行へ進む

590: 検索と表示のときは氏名の入力を求め、次の行へ進む

610–650: 索引ファイルと住所録ファイルを開く処理

660–690: 索引を調べ、登録されているデータを読み込む処理

700: 全データ表示なら 730行へジャンプする

710–720: 読み込んだデータに入力された氏名があるかどうか調べ、なければ 800行へ

730: 氏名を含むデータを画面に表示する

```
740 IF N=5 THEN 770
750 IF INKEY$="" THEN 750
760 A$=INKEY$:GOTO 800
770 INPUT "削除 1:Ok/0:NG";A
$
780 IF A$="１" OR A$="1" THE
N MID$(FT$,C1,1)="0"
790 PUT #1,1:CLOSE:GOTO 140
800 NEXT C1
810 CLOSE
820 PRINT "        DATA END "
830 PRINT "    HIT ANY KEY !
"
840 IF INKEY$="" THEN 840
850 A$=INKEY$:GOTO 140
860 CALL CLS:END
```

- 740：削除するときは 770行へジャンプする
- 760：何かキーが押されたら 800行へ
- 770：削除してよいか確認を求める
- 780：削除してよければ、索引のそのデータにあたる部分を「０」にする
- 790：索引ファイルを書き換え、両方のファイルを閉じて140行へ戻る
- 810〜850：すべてのデータを表示し終わったときの処理
- 860：終了の処理

第1章

「MSX-DOS2」の紹介



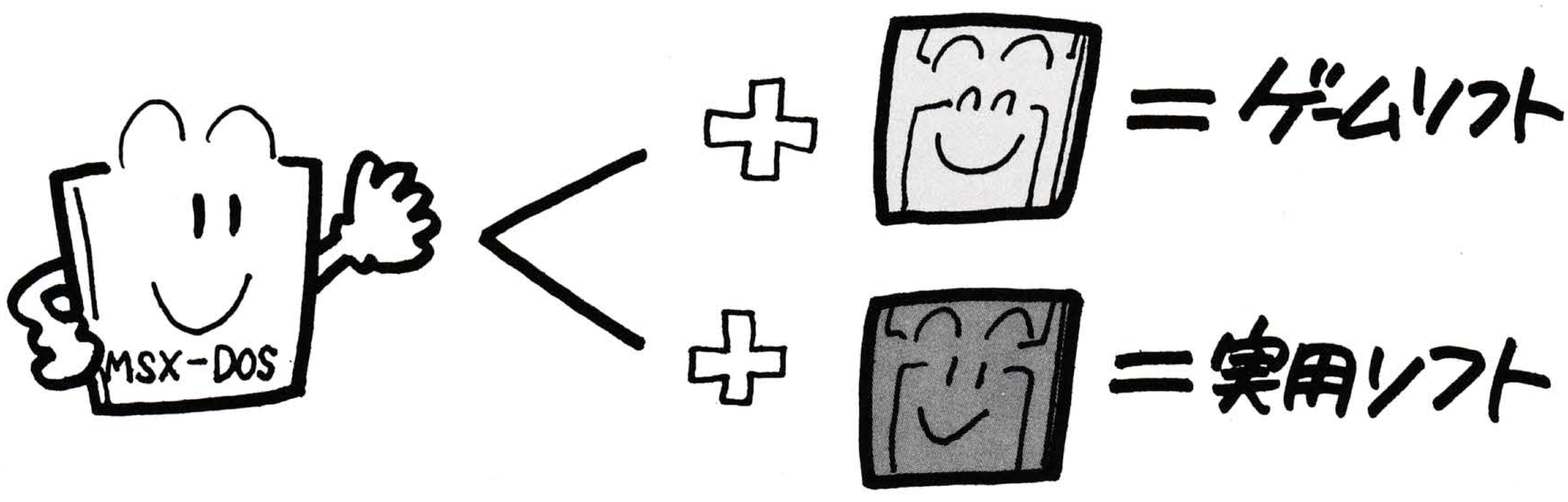
「MSX-DOS2」って何だろう?

ＭＳＸ turbo Ｒパソコンには、「ＭＳＸ－ＤＯＳ２」と書かれたフロッピーディスクが付属しています。ＭＳＸは分かりますが、「ＤＯＳ」っていったい何なのでしょうか?

ＢＡＳＩＣはできるだけいろいろな用途に使えるようにたくさんの機能をもっているため、パソコンのメモリの半分以上をＢＡＳＩＣインタープリタが使っています。
しかし、パソコンを動かすためのプログラムは、何もＢＡＳＩＣでなければいけないわけではありません。実際、パソコンの付属ソフトや市販ソフトなどは、さまざまなプログラミング言語で書かれたあと、機械語に書き直した状態でフロッピーディスクに保存されて供給されています。

また、多くの市販ソフトは限られた目的のために作られていますから、そんなにいろいろな機能は必要ありません。それよりは自分自身のプログラムやデータの領域として使うほうが、メモリを有効に使えることになります。

そこで、フロッピーディスクに保存された機械語ソフトをより有効に使用できるように、フロッピーディスクドライブ内のファイルの操作に必要な機能だけを集めたものが**「ＤＯＳ」**（ドス：Disk Operating System）です。
「ＭＳＸ－ＤＯＳ２」は、ＭＳＸ turbo Ｒパソコン用のＤＯＳで、従来のＭＳＸ、ＭＳＸ２、ＭＳＸ２＋パソコン用のＤＯＳである「ＭＳＸ－ＤＯＳ」よりも強力な機能をもっています。



主な変更点は次のとおりです。
・「サブディレクトリ」の採用により、ファイルを階層化して管理できるようになった。
・ＲＡＭディスク機能が強化された。
・漢字ＢＡＳＩＣに対応して、ＭＳＸ－ＤＯＳ２も漢字モードで使用できるようになった。
・ＭＳＸ turbo Ｒに対応して、高速モードで動作できるようになった。

MSX-DOS2の用途

市販ソフトの中には、ＭＳＸ－ＤＯＳ２や、従来のＭＳＸ－ＤＯＳが組み込まれているものがたくさんあります。これらのソフトを使うときは、使う人がＤＯＳのことを何も知らなくてもまったく問題ありません。

ここでは、ＭＳＸ－ＤＯＳ２の使いかたを知っていると便利な場合をあげます。

・フロッピーディスクに保存したファイルを操作するとき
ファイルの一覧を見たり、たくさんのファイルを他のディスクにコピーしたりするときには、ＢＡＳＩＣよりＭＳＸ－ＤＯＳ２を使うほうが便利です。

・ＢＡＳＩＣ以外の言語でプログラムを自作するとき
この場合、ＭＳＸ－ＤＯＳ２で動作する「コンパイラ」や「エディタ」などという機械語ソフトが必要です。

従来のMSX-DOSとの互換性

ＭＳＸ－ＤＯＳ２は、従来のＭＳＸ－ＤＯＳの命令や機能をそのまま含み、さらに新しい命令や機能を追加したものです。

ＭＳＸ turbo Ｒ パソコンは、ＭＳＸ－ＤＯＳ２、ＭＳＸ－ＤＯＳのどちらでもスタートすることができます。

・ＭＳＸ－ＤＯＳ２をスタートしたときには、本体の動作速度が「高速モード」になります。
　したがって、ＭＳＸ－ＤＯＳ２が組み込まれている市販ソフトは、高速モードで動作します。
・ＭＳＸ－ＤＯＳをスタートしたときには、本体の動作速度が「標準モード」になります。
　したがって、従来のＭＳＸ－ＤＯＳが組み込まれている市販ソフトは、標準モードで動作します。

ＭＳＸ－ＤＯＳ２のフロッピーディスクのフォーマットは、ＤＩＳＫ　ＢＡＳＩＣ２と共通になっています。

・ＤＩＳＫ　ＢＡＳＩＣ２とＭＳＸ－ＤＯＳ２は、互いにフォーマットしたフロッピーディスクを交換して使用することができます。その場合、互いに使用する機能を制限されません。
・ＭＳＸ－ＤＯＳ２でフォーマットしたフロッピーディスクをＭＳＸ－ＤＯＳまたはＤＩＳＫ　ＢＡＳＩＣで使用するときは、保存されているファイルを使用できない場合があります。
・ＭＳＸ－ＤＯＳまたはＤＩＳＫ　ＢＡＳＩＣでフォーマットしたフロッピーディスクをＭＳＸ－ＤＯＳ２やＤＩＳＫ　ＢＡＳＩＣ２で使用するときは、使用する機能を一部制限されます。

第2章

BASICとはひと味ちがうMSX－DOS2。この強力さが快感です。

MSX-DOS2の使いかた



MSX-DOS2を使ってみよう

では、MSX－DOS 2の使いかたをかんたんに説明します。

MSX-DOS2のスタートのしかた

以降の操作は、パソコンの機種により異なる場合があります。同じ方法でスタートできないときは、お使いの機種の取扱説明書をお読みください。

① パソコンの内蔵ソフト切り替えスイッチを「切」にします。
② パソコンに付属しているシステムディスク2をドライブAに入れます。
③ パソコンのリセットボタンを押すか、または電源スイッチを入れ直します。
④ 付属ソフトのメニュー画面が表示されます。
⑤ 「DOS 2」を選ぶ数字キーを押します。
⑥ 画面に下の文字が表示されます。これでMSX－DOS 2がスタートしています。

```
A>█
```

システムディスクの基礎知識

本体に付属しているシステムディスクには、MSX－DOS 2以外にも各種の付属ソフトが入っています。付属ソフトを誤った書き換えなどの事故から保護するためにも、内容をすべて別のディスクにコピーして、日常はそのディスクを「システムディスク」として使うようにしましょう。

システムディスクを作るときは、必ず次のようにします。

① MSX－DOS 2をコピーする新しいディスクをドライブに入れます。
② キーボードから「**DIR　A：**↵ 」と入力し、消えては困るファイルが保存されていないことを確かめます。
③ 「**FORMAT　A：**↵ 」と入力します。
④ [2] を押します。
⑤ ↵ を押します。
フォーマットが始まります。フォーマットには約2分間必要です。
フォーマットが済んだ新しいディスクを取り出し、付属のシステムディスク2をドライブに入れます。

メモ： ②の操作をしたときに「ドライブA:用のディスクを入れて何かキーを押してください」と表示されたときは、そのまま ↵ を押してください。
：フォーマットされていないディスクをドライブに入れて②の操作をしたときには、「ディスクはフォーマットされていません」と表示されます。

⑥ 「ＤＩＳＫＣＯＰＹ　Ａ：　Ｂ：／Ｓ ⏎ 」と入力します。
画面に次のように表示されます。

```
複写元ディスクをドライブA:
複写先ディスクをドライブB:
何かキーを押して下さい... █
```

⑦ ⏎ を押します。
画面に次のように表示されます。

```
ドライブB:用のディスクを入れ
て
何かキーを押して下さい█
```

⑧ 付属のシステムディスク２を取り出し、フォーマットしたディスクをドライブに入れ、⏎ を押します。
画面に次のように表示されます。

```
ドライブA:用のディスクを入れ
て
何かキーを押して下さい█
```

⑨ 付属のシステムディスク２をドライブに入れます。

⑩ 画面に下のように表示されるまで、⑦〜⑨の操作を繰り返します。

```
コピーが終了しました
他のディスクをコピーしますか
(Y/N)? █
```

⑪ N を押す。

⑫ ⏎ を押します。

これでＭＳＸ−ＤＯＳ２のシステムディスクができました。

ご注意：「ＤＩＳＫＣＯＰＹ」コマンドは、システムディスクからプログラムを読み込んで動作します。必ずシステムディスクをドライブに入れておいてください。

DOS編

MSX-DOS2の基礎知識

「A＞」

ＢＡＳＩＣのダイレクトモードでは、パソコンが命令の入力を待っているとき、画面に「Ｏｋ」と表示されました。
ＭＳＸ－ＤＯＳ２でこれにあたるのが、**「Ａ＞」（Ａプロンプト）**です。
左端の「Ａ」は、ドライブ名を省略して命令したときにＭＳＸ－ＤＯＳ２が操作するフロッピーディスクドライブ（この場合、Ａドライブ）をあらわします。

「コマンドライン」

ＢＡＳＩＣでは「Ｏｋ」の下にカーソルが表示されましたが、ＭＳＸ－ＤＯＳ２では「＞」の右にカーソルが表示されます。

Ａ＞■

現在カーソルが表示されている「＞」がある行を**「コマンドライン」**と呼びます。
ＭＳＸ－ＤＯＳ２では、命令の入力をすべて「コマンドライン」で行います。

「日本語モード」と「英文モード」

ＭＳＸ－ＤＯＳ２には、「日本語モード」と「英文モード」の２つのモードがあります。
「日本語モード」では、表示されるメッセージが日本語になり、漢字ＢＡＳＩＣなどで作成した漢字のファイル名が正常に表示されます。
ＭＳＸ－ＤＯＳ２を初めてスタートしたときは「日本語モード」になっています。
「英文モード」で使用するときは、「ＫＭＯＤＥ　ＯＦＦ」と入力してください。
「ＫＭＯＤＥ　０」と入力すると、元の日本語モードに戻ります。

ＭＳＸ－ＤＯＳ２からＢＡＳＩＣをスタートさせる　（ＢＡＳＩＣ）

ＭＳＸ－ＤＯＳ２には、**「ＢＡＳＩＣ」**という命令があります。これは、ＭＳＸ－ＤＯＳ２からＤＩＳＫ　ＢＡＳＩＣをスタートさせる命令です。
ＢＡＳＩＣで**「ＣＡＬＬ　ＳＹＳＴＥＭ」**（コール・システム）という命令を実行させることにより、再びＭＳＸ－ＤＯＳ２に戻ることができます。
（このときは、ＡドライブにＭＳＸ－ＤＯＳ２のシステムディスクが入っていることが必要です）

ファイルとサブディレクトリの一覧を表示する　（ＤＩＲ）

ＢＡＳＩＣで、フロッピーディスクに保存されているファイルの一覧を見るには、ＦＩＬＥＳ命令を使いました。ＭＳＸ－ＤＯＳ２でそれにあたるのが、**「ＤＩＲ」**（ディレクトリ）命令です。

ＤＩＲ　ドライブ名　ファイルスペック

例）　ＤＩＲ

ＤＩＲ　COMMANDＤ２．COM

ＭＳＸ－ＤＯＳ２のファイルの一覧は、下のような形式になります。

・日本語モードのとき（29桁表示：スタートしたときの状態）

ＦＩＬＥＮＡＭＥ	ＥＸＰ	１２３４
（ファイル名）	（拡張子）	（ファイルの大きさ）

・英文モードのとき（29桁～38桁表示のとき）

ＦＩＬＥＮＡＭＥ	ＥＸＰ	１２３４	９９－０９－０９
（ファイル名）	（拡張子）	（大きさ）	（保存された日付け）

・英文モードのとき（39桁以上の表示のとき）

ＦＩＬＥＮＡＭＥ	ＥＸＰ	１２３４	９９－０９－０９	３：１２ｐ
（ファイル名）	（拡張子）	（大きさ）	（日付け）	（保存された時刻）

メモ：日本語モードと英文モードのどちらでも、画面の１行に表示される文字の桁数を、一定の範囲で調節することができます。それには**「ＭＯＤＥ」**（モード）コマンドを使用します。上の「英文モードのとき」の80桁表示を見るときは、「ＫＭＯＤＥ　ＯＦＦ」を入力して英文モードにし、「ＭＯＤＥ　８０」と入力してから「ＤＩＲ」と入力します。

ＤＯＳ編

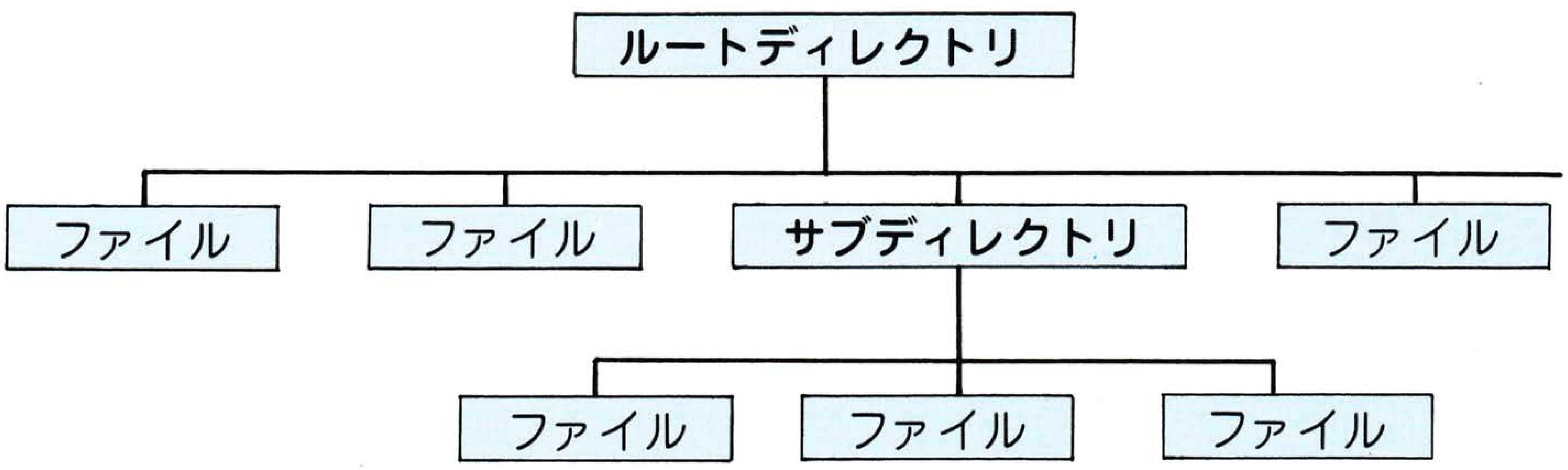
ルートディレクトリとサブディレクトリ

「ＤＩＲ」命令で表示されるファイルの一覧表示を「ディレクトリ」と呼びます。
先ほど表示されたディレクトリをよく見ると､｢ファイル容量｣の位置に､数字ではなく「＜ｄｉｒ＞」という文字が表示されているものがあります。

```
UTILS     <dir>
HELP      <dir>
KHELP     <dir>
```

これらは、実はファイルではなく、別のディレクトリなのです。ＭＳＸ－ＤＯＳ２では、１枚のディスクに対して１つのディレクトリではなく、２つ以上のディレクトリを作ることができ、必要に応じて、操作するディレクトリを指定することができます。

ＭＳＸ－ＤＯＳ２でフォーマット（初期化）したばかりのディスクには、ディレクトリは１つしかありません。このディレクトリを「ルートディレクトリ」と呼びます。他のディレクトリは、すべて「サブディレクトリ」と呼ばれ、後述するサブディレクトリを作る命令（ＭＫＤＩＲ命令）によって現在のディレクトリの下に作られます。



先ほどのディレクトリで「＜ｄｉｒ＞」と表示された行は、ルートディレクトリの下に作られているサブディレクトリの情報を表示するものです。

ＵＴＩＬＳ　　　　＜ｄｉｒ＞　××－××－××

ディレクトリ名　　　　サブディレクトリが作られた日付け

サブディレクトリは、ファイルと同じように名前を付けて操作します。この名前を「ディレクトリ名」と呼びます。（ルートディレクトリは、名前の代わりに「￥」記号を使います。）

　例）ＤＩＲ　ＵＴＩＬＳ

サブディレクトリは、さらにその下に別のサブディレクトリを作ることができます。
サブディレクトリにあるファイルを操作するときは、ファイル名の前に、そのファイルがあるところまでのすべてのサブディレクトリのディレクトリ名を「￥」でつないで入力します。これを「パスリスト」と呼びます。

　例）ＤＩＲ　ＵＴＩＬＳ￥ＵＮＤＥＬ．ＣＯＭ

ディレクトリ名やパスリストを省略して操作できるディレクトリを「カレントディレクトリ」と呼びます。最初はルートディレクトリがカレントディレクトリになっていますが、カレントディレクトリを変更する命令（ＣＨＤＩＲ命令）により変更できます。

サブディレクトリ関係の命令

MKDIR　……　新しくサブディレクトリを作る
CHDIR　……　カレントディレクトリを変更する
RMDIR　……　ファイルやディレクトリのない空のサブディレクトリを削除する

命令の入力のしかた

DIR命令の例でも分かるように、MSX－DOS2ではBASICと命令の入力のしかたが少し異なります。

・ファイルスペックは「″」で囲まずに入力します。
・命令とファイルスペックなど、入力する各項目のあいだには、必ず1文字以上の空白を入れます。ただし、ドライブ名とファイル名、パスリスト、ファイル名と「.」を含めた拡張子のあいだ（つまりファイルスペックの中）には空白を入れてはいけません。
・1つのコマンドラインで2つ以上の命令を入力することはできません。
・BASICのプログラムモードにあたるものはありませんが、そのかわり、決まった手順で行う命令を登録する**「バッチファイル」**というファイルを作ることができます。バッチファイルはコマンドラインにファイル名までを入力するだけで実行されます。

ファイルをコピーする　（COPY）

カレントディレクトリに保存されているファイルを他のディスクにコピーするときは、次のようにします。（例として、ルートディレクトリの「COMMAND2.COM」をコピーします）

●**ドライブ1台でコピーするとき**（ドライブ2台でコピーするときは、146ページ）

① フォーマット済みの書き込み可能なフロッピーディスクを用意します。
② MSX－DOS2をスタートします。
③ コピーする元のライトプロテクトタブを「書き込み禁止」の位置にセットし、ドライブに入れます。
④ コマンドラインに次のように入力します。（⇨メモ）

A>COPY　COMMAND2.COM　B:　↵

⑤ アクセスランプ（またはIN　USEランプ）が点灯し、「COMMAND2.COM」がパソコンのメモリに呼び出されます。

メモ：サブディレクトリを使ったコピー
パスリストを指定することにより、サブディレクトリにファイルをコピーできます。
例）　A>COPY　COMMAND2.COM　B:HOBBY¥PET
　　　A>COPY　A:ZOO¥DOG　B:HOBBY¥CAT

DOS編

⑥　次のように画面に表示されます。

```
ドライブB:用のディスクを入れ
て
何かキーを押して下さい▮
```

⑦　アクセスランプ（またはＩＮ　ＵＳＥランプ）が消えてから、コピー元のディスクと用意したディスクを入れ替え、⏎ を押します。

⑧　アクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯し、ＣＯＭＭＡＮＤ２．ＣＯＭが保存されます。一度にコピーできたときは⑫の画面になります。

⑨　ファイルが大きすぎて一度にコピーできないときは、画面に次のように表示されます。

```
ドライブA:用のディスクを入れ
て
何かキーを押して下さい▮
```

⑩　ランプが消えてから、再びディスクを入れ替え、⏎ を押します。

⑪　画面の表示にしたがい、⑥～⑩の操作を繰り返します。

⑫　コピーが終わると、画面に下のように表示されます。

```
A>▮
```

●ドライブ２台でコピーするとき

メモ：ドライブ１台内蔵のパソコンにドライブをもう１台増設したときは、増設したドライブが「ドライブＡ」、内蔵ドライブが「ドライブＣ」になります。
：ドライブ２台内蔵のパソコンでは、片方が「ドライブＡ」、もう一方が「ドライブＢ」になります。どちらがＡ，Ｂかは、本体に表示されています。
：ドライブ１台内蔵のパソコンにドライブをもう１台増設して、コントロール立ち上げをしたときは、増設ドライブが「ドライブＡ」、内蔵ドライブが「ドライブＢ」になります。

①　フォーマット済みの書き込み可能なフロッピーディスクを用意します。

②　ＭＳＸ－ＤＯＳ２をスタートします。

③　コピーする元のディスクのライトプロテクトタブを使用禁止の位置にセットし、ドライブＡに入れます。

④　用意したディスクをドライブＣ（またはドライブＢ）に入れます。

⑤コマンドラインに次のように入力します。

A>COPY　COMMAND2．COM　C：↵

ドライブを増設しているとき

または、

A>COPY　COMMAND2．COM　B：↵

ドライブ2台内蔵、またはドライブを増設してコントロール立ち上げをしたとき

⑥ ドライブAのアクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯し、COMMAND2．COMがパソコンのメモリに呼び出されます。
⑦ ドライブAのアクセスランプ（またはＩＮ　ＵＳＥランプ）が消え、かわってドライブC（またはB）のアクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯してCOMMAND2．COMが保存されます。
⑧ ファイルが大きすぎて一度にコピーできないときは、⑥と⑦の動作が繰り返されます。
⑨ コピーが終わると、画面に下のように表示されます。

```
A>▮
```

（サブディレクトリを使ったコピー⇨145ページ）

ディスクをコピーする

1枚のフロッピーディスクに保存されているすべてのファイルを他のディスクにコピーするときは、次のようにします。（ドライブ1台の場合）

① フォーマット済みの書き込み可能なフロッピーディスクを用意します。
② MSX−DOS2をスタートします。
③ システムディスクをドライブに入れた状態で、次のように入力します。

A>DISKCOPY　A：　B：↵

④ 画面に次のように表示されます。

```
複写元ディスクをドライブA:
複写先ディスクをドライブB:
何かキーを押して下さい... ▮
```

⑤ コピーする元のディスクのライトプロテクトタブを「書き込み禁止」位置にセットしてドライブに入れ、↵ を押します。
⑥ 画面に次のように表示されます。

```
ドライブB:用のディスクを入れ
て
何かキーを押して下さい█
```

⑦ アクセスランプ（またはＩＮ　ＵＳＥランプ）が消えてから、コピー元のディスクと用意したディスクを入れ替え、↵ を押します。
⑧ アクセスランプ（またはＩＮ　ＵＳＥランプ）が点灯し、メモリに呼び出されたファイルが次々に保存されます。
⑨ 画面に次のように表示されます。

```
ドライブA:用のディスクを入れ
て
何かキーを押して下さい█
```

⑩ ランプが消えてから、再びディスクを入れ替え、↵ を押します。
⑪ 画面に下のように表示されるまで、⑤〜⑩の操作を繰り返します。

```
コピーが終了しました
他のディスクをコピーしますか
(Y/N)? █
```

⑫ N を押す。
⑬ システムディスクをドライブに入れ、↵ を押します。

メモ：ドライブ２台でコピーするとき

・ドライブ名については、146ページのメモを参照してください。
・コピー元のディスクをドライブＡ、用意したデイスクを他方のドライブに入れます。
・コマンドラインに次のように入力します。

Ａ>ＤＩＳＫＣＯＰＹ　Ａ：　Ｃ：……ドライブを１台増設しているとき
Ａ>ＤＩＳＫＣＯＰＹ　Ａ：　Ｂ：……ドライブを２台内蔵、またはコントロール立ち上げをしたとき

・メッセージが出てから ↵ を押すと、コピーが始まります。終了のメッセージ（⑪のメッセージ）が表示されるまで、ディスクの出し入れはしないでください。
・終了のメッセージが表示されたら、N を押します。

ワイルドカード

「＊」（アスタリスク）は、ファイルスペックの代わりに入力したときは、文字数の決まっていない不特定の文字列をあらわし**「ワイルドカード」**と呼ばれています。
ワイルドカードに使える文字には「＊」の他にも**「？」**（クエスチョン）があり、ファイルスペックの中で使うときには、不特定の１文字をあらわします。
例えば、「＊．＊」はすべてのファイルをあらわし、「ＰＲＯＧＲＡＭ？」は、「ＰＲＯＧＲＡＭ」の後ろに何か１文字続くファイル名をもつすべてのファイルをあらわします。

ＭＳＸ－ＤＯＳ２のディスクと従来のＭＳＸ－ＤＯＳのディスクの見分けかた

ＭＳＸ－ＤＯＳ２の命令や機能の一部は、ＭＳＸ－ＤＯＳのディスクに対しては使用できません。あるディスクがＭＳＸ－ＤＯＳ２のディスクかＭＳＸ－ＤＯＳのディスクかが分からなくなったときは次の命令を使って調べてください。

```
A>CHKDSK B:
```

入力すると、システムディスクから「ＣＨＫＤＳＫ」のプログラムを呼び出して、ドライブBにディスクを入れることを求める表示がでます。ディスクを交換して ↵ キーを押してください。

ＭＳＸ－ＤＯＳ２のディスクのとき‥‥ディスクの合計容量、保存されているファイル数、サブディレクトリ数、ディスクの空き容量が表示されます。

ＭＳＸ－ＤＯＳのディスクのとき‥‥‥「ディスクはMSX-DOS2でフォーマットされていません」と表示されます。

操作が分からなくなったときは

MSX－DOS2を操作中に、命令の使いかたや機能が分からなくなったときは、次のように入力します。

A>HELP　（命令）

例）A>HELP　DIR

サブディレクトリ「KHELP」にある説明文のファイル「DIR．HLP」の内容が画面に表示されます。

画面に表示される文字数（桁数または行数）を増やしたいとき
画面の表示を高速にしたいとき（漢字モードと英数・かなモード）

MSX－DOS2がスタートしたときには、1画面は29桁（半角）×13行に設定されています。

画面に表示できる文字の桁数・行数を増やしたいときは次のように入力します。

A>KMODE　（漢字モード番号）

例）A>KMODE　0　……　29桁×13行（スタート時と同じ）
　　A>KMODE　1　……　36桁×13行
　　A>KMODE　2　……　29桁×24行（インターレース表示）
　　A>KMODE　3　……　36桁×24行（インターレース表示）

※インターレース表示は、画面のチラツキが多くなるため、お勧めできません。

表示が英文になってもよいから画面の表示をもっと高速にしたいというときは、MSX－DOS2を英数・かなモードにすることにより、表示される速度を少し速くすることができます。

A>KMODE　OFF

「KMODE　（漢字モード番号）」と入力することにより、漢字モードに戻ることができます。

こんなメッセージが表示されたときは

MSX－DOS2は、BASICとは異なるエラーメッセージを表示します。
その中で、次のような表示が出たときは、指示を入力することが必要です。

中止（A）／再試行（R）／無視（I）？ ■

Abort, Retry, Ignore? ■

エラーが発生した操作（またはバッチファイルの実行）を、

・中止するとき　……　[A] キーを押す。
・再実行するとき　……　[R] キーを押す。
・無視して先に進むとき……　[I] キーを押す。

※ [I] キーを押して処理を続けると、ファイルやディスクを壊すことがあります。通常は使用しないでください。

資　料　編

漢字ＢＡＳＩＣの規則概説　……………………　152

キャラクターコード表　……………………　174

ＢＡＳＩＣリファレンス表　……………………　175
（ＤＩＳＫ　ＢＡＳＩＣを含む）

ＢＡＳＩＣ予約語表　……………………　194

コントロールコード表　……………………　195

メモリマップ（ＢＡＳＩＣ使用時）　……………　196

メモリマップ（ＭＳＸ－ＤＯＳ２使用時）　………　198

ＭＳＸ－ＤＯＳ２リファレンス表　……………　199

ＢＡＳＩＣエラーメッセージ表　………………　207

ＭＳＸ－ＤＯＳ２エラーメッセージ表　…………　209

ローマ字・かな変換表　……………………　212

命令索引　……………………………　214

推奨図書一覧　…………………………　218

漢字コード表の補足説明　……………………　219

漢字BASICの規則概説

これからのページでは、「漢字ＢＡＳＩＣ」（およびＤＩＳＫ　ＢＡＳＩＣ２）の一般的規則の概略を説明しています。必要なときにお読みください。

※ＢＡＳＩＣは初めてという人は、いきなりこの「規則概説」を読んでも理解しにくいかもしれません。まずこの本の前のほう（「ＢＡＳＩＣ編」）で説明されている規則を理解するようにしてください。特に複雑なプログラム以外は、それだけで十分に作れます。ＢＡＳＩＣでのプログラム作成になれてくれば、ここに書いてあることも自然に理解できるようになるでしょう。

漢字BASICで扱えるデータ(定数・変数)

分類		内容	型宣言文字（＊1）
文字型		255文字以内の文字列	$ 例　A＄＝″MOJI″
数値型	整数	−32768～＋32767の整数 （小数点以下の値は切り捨て） 表示形式（＊4）	％ 例　12.99％→12 A％＝123
	単精度実数	有効数字６桁の精度の実数 指数範囲は 10^{-64} ～ 10^{+62} 指数定数の表記（＊2）	！　（＊3） 例　12345678! →12345700 A！＝1.2345
	倍精度実数	有効数字14桁の精度の実数 指数範囲は 10^{-64} ～ 10^{+62} 指数定数の表記（＊2）	＃　（＊3） 例　1.23456789♯ A＃＝98.765432

＊１）**型宣言文字**　変数名や定数の後ろに付けることにより、そのデータの型を宣言（定義）する文字。型宣言文字を付けていない定数および変数（ＤＥＦＳＴＲ，ＤＥＦＩＮＴ，ＤＥＦＳＮＧ，ＤＥＦＤＢＬ文により定義されているものを除く）は、倍精度型実数とみなされます。

＊２）**指数定数の表記**　**（数値）Ｅ±ｎ**　ｎは指数（−６４～＋６２）
例　１２３４５６．７８＝１．２３４５６７８Ｅ＋５　　０．００００００３４＝３．４Ｅ−６

＊３）型宣言による**有効桁を超える数値**は、有効桁の次の桁で四捨五入されます。

例

定義式の例	取り扱われる数値
Ａ！＝１２３４．５６７８９０１２３４	１２３４．５７
Ａ＃＝１２３４．５６７８９０１２３４５６	１２３４．５６７８９０１２３５

＊４）　**整数の表示形式**
整数は２バイトで、各ビットに下記のような「重み」を付けたビット配列であらわします。

２バイト目（上位）								１バイト目（下位）							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

１６進形式　　先頭に&Hを付けた０～９，Ａ～Ｆの並び。２進形式のビット配列を４ビットずつ区切ってあらわす。
（例　&Ｂ１１１１１０１００１０１１１０１＝&ＨＦＡ５Ｄ）
１０進形式の　１０，１１，１２，１３，１４，１５　をそれぞれ　Ａ，Ｂ，Ｃ，Ｄ，Ｅ，Ｆ　に対応させてあらわす。

変　数

変数とは、プログラム中で使う文字列や数値をしまっておくことができる「入れ物」のことで、各変数にはそれぞれメモリの特定の領域が割り当てられます。変数には区別のため、それぞれ変数名を付けます。

変数名
・２文字までの文字。（１文字目は英字、２文字目は英字または数字）３文字目以降は無視。ただし、型宣言文字は変数名の文字数として数えない。
・英字の大文字、小文字は区別しない。（ＡＢとａｂは同じ変数となる）
・予約語を含まないこと。（⇨194ページ「予約語表」）

配列変数

配列変数とは添え字により区別されたいくつかの要素から構成された変数です。

例　AA（L，M，N）

- 変数名：AA
- 各次元の要素数（各要素は 0 から始まる）：L，M，N
- 配列の次元数（上の例は 3 次元）

配列変数は、使用する前にＤＩＭ文で配列の大きさ（次元数と各次元の要素数の上限）を宣言して使用します。

例　DIM　A（10，20，5）

ただし、3 次元までの配列変数は宣言なしでも使用できます。この場合、各次元の要素数の上限は、自動的に10まで（0 を含め11個）に設定されます。

演算子・関係演算子・論理演算子

ＢＡＳＩＣで使用する演算子（計算に用いる記号）は、次のようになっています。

算術演算子	指数	負号	×，÷	整数除算	剰余	＋，－
	∧	－	＊，／	￥	MOD	＋，－

関係演算子	等号	より大	より小	以上	以下	不等号
	＝	＜	＞	＜＝	＞＝	＜＞
				＝＜	＝＞	＞＜

※関係演算子は 2 つの数値または文字列を比べるのに用います。演算子がしめす関係が、正しければ結果は真（－1）に、正しくなければ偽（0）になります。

論理演算子	否定	論理積	論理和	排他的論理和	包含	同値
	NOT	AND	OR	XOR	IMP	EQV

※論理演算子は複数の条件を調べたり、データをビット操作したりするときなどに用います。

例） 4ビットのデータ2個のビット操作例

入力条件	X	0	1	0	1
	Y	0	0	1	1
NOT X		1	0	1	0
X AND Y		0	0	0	1
X OR Y		0	1	1	1
X XOR Y		0	1	1	0
X IMP Y		1	0	1	1
X EQV Y		1	0	0	1

操作した結果（NOT X〜X EQV Y）

演算の優先順位

・ひとつの式の中では、各表の左側の演算子から先に計算される。
・表が異なる演算子のあいだでは、算術演算子・関係演算子・論理演算子の順に計算される。
・式の中に（　）がある場合は、内側の（　）の中から先に計算される。

漢字BASICの基本モード

漢字ＢＡＳＩＣは次のような構成になっています。

漢字ＢＡＳＩＣ

英数・かなモード	**漢字モード**
テキスト画面 （スクリーンモード０，１）	**漢字テキスト画面** （スクリーンモード０，１）
グラフィック画面 （スクリーンモード２～12）	**漢字グラフィック画面** （スクリーンモード２～12）

２つの基本モードは、次の命令で切り替えます。

英数・かなモード	→	漢字モード	……	ＣＡＬＬ　ＫＡＮＪＩ
漢字モード	→	英数・かなモード	……	ＣＡＬＬ　ＡＮＫ

英数・かなモードの特徴

・従来のＭＳＸ２　ＢＡＳＩＣで作成したプログラムが変更なしで実行できる。

漢字モードの特徴

・漢字テキスト画面に漢字などの全角文字を表示できる。
・漢字グラフィック画面に漢字を表示するときでも、ＬＯＣＡＴＥ文やＰＲＩＮＴ文が使用できる。
・かな漢字変換機能がつき、キーボードからかなを入力して漢字に変換することができる。
・ＢＡＳＩＣのプログラム中の文字列やＲＥＭ文などに漢字を直接使用できる。
・漢字などの全角文字の文字列をあつかう命令や関数が追加されている。

画面モード

<table>
<tr><th>スクリーンNo.</th><th>モード</th><th>最大表示桁数または解像度</th><th>表示色数</th><th>色指定</th><th>スプライト</th></tr>
<tr><td>0</td><td rowspan="2">テキストモード</td><td>80桁×24行</td><td rowspan="2">16パレット中の2色</td><td rowspan="2">前景色
背景色
周辺色</td><td>表示できない</td></tr>
<tr><td>1</td><td>32桁×24行</td><td rowspan="3">8×8ドット
1ラインに4個まで同時表示可能</td></tr>
<tr><td>2</td><td>高解像度グラフィックモード</td><td>256×192ドット</td><td rowspan="4">16パレット</td><td>8ドットを1単位とする領域ごとに2色</td></tr>
<tr><td>3</td><td>マルチカラーモード</td><td>64×48ブロック</td><td>1ブロック毎に指定</td></tr>
<tr><td>4</td><td>高解像度グラフィックモード</td><td>256×192ドット</td><td>8ドットを1単位とする領域ごとに2色</td><td rowspan="8">8×8ドットまたは16×16ドット
1ラインに8個まで同時表示可能</td></tr>
<tr><td>5</td><td rowspan="4">ビットマップグラフィックモード</td><td>256×212ドット</td><td rowspan="4">1ドット毎に指定</td></tr>
<tr><td>6</td><td rowspan="2">512×212ドット</td><td>4パレット</td></tr>
<tr><td>7</td><td>16パレット</td></tr>
<tr><td>8</td><td rowspan="4">256×212ドット</td><td>256色（固定）</td></tr>
<tr><td>10</td><td rowspan="3">自然画表示モード</td><td rowspan="2">パレット・YJK混在
16パレット
YJK表示時、
12，499色</td><td rowspan="2">パレット表示は、
1ドット毎に指定
YJK表示は、連続した4ドット毎に指定</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td><td>YJK表示のみ
19，268色</td><td>連続した4ドット毎に指定</td></tr>
</table>

※パレットは、赤緑青各8段階の組み合わせからなる 512色中よりパレットごとに色指定することができる。
※1ブロックは、4×4ドットで構成。
※自然画は、カラー写真などを特殊な装置を使って画像データに変換した絵のこと。
※ＹＪＫは、従来のＲＧＢ方式とは異なる自然画表示のための色表現方式。（⇨161ページ）
ＢＡＳＩＣの命令では直接使用できない。

英数・かなモードの文字表示

画面モード	ANKサイズ	表示命令	位置指定命令	漢字サイズ	表示命令
スクリーン0	6×8ドット	PRINT	LOCATE	表示できない	
スクリーン1	8×8ドット	PRINT	LOCATE	表示できない	—
スクリーン2	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン3	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン4	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン5	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン6	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン7	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン8	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン10	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン11	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI
スクリーン12	8×8ドット	OPEN，PRINT＃	最終参照点	16×16ドット	PUT KANJI

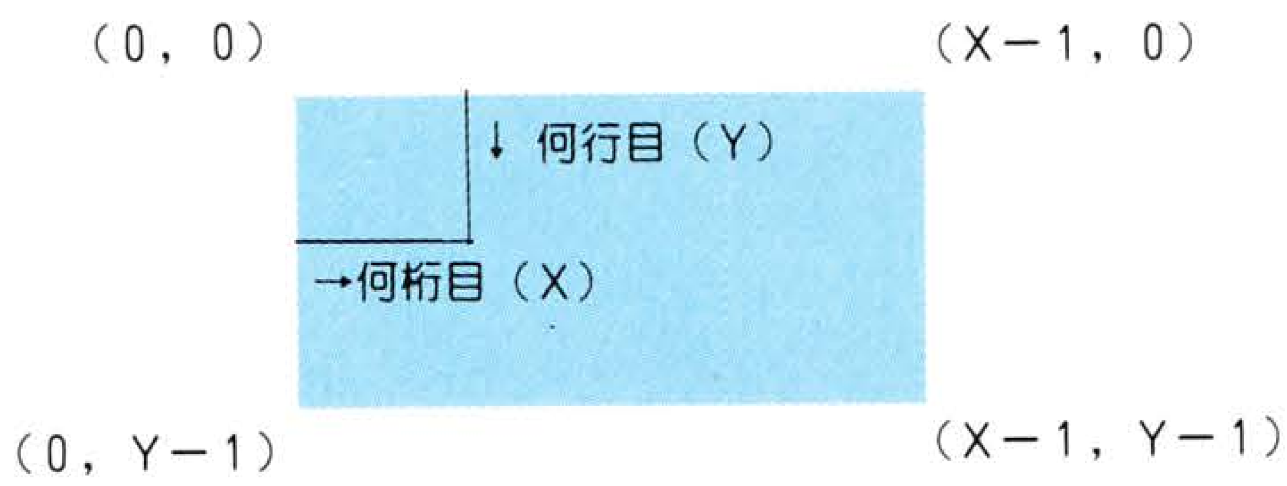
※「ANK」は、アルファベット・数字・カタカナ・ひらがな・グラフィック文字およびそれらの記号などの、キーボードから直接入力できる文字の総称。
※スクリーン11〜12は、表示色の設定方法が他の画面と異なるため、文字表示に適さない。
※最終参照点は、最後に実行したグラフィック関係の命令で指定した位置座標のこと。通常は、PRESET命令などを使って設定する。

スクリーン0の文字表示

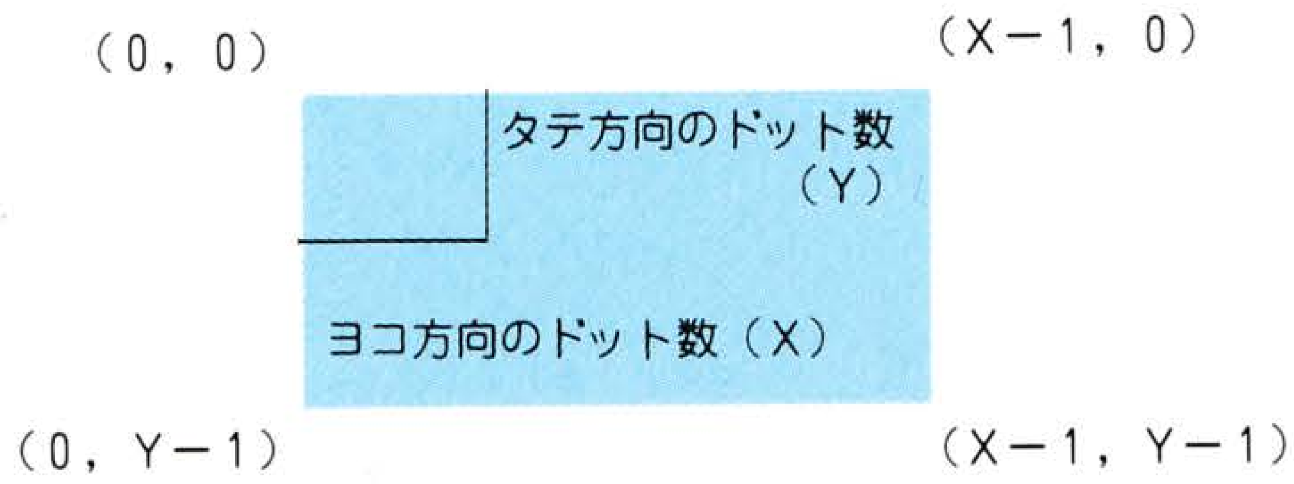
・スクリーン0で1行が41桁以上のときは、画面のヨコ方向のドット密度が2倍になるため、文字の幅が半分になる。
・スクリーン0では文字の表示パターンの右2列分のドットをカットして表示するため、ひらがなやグラフィック文字の一部は文字の右端が欠けて表示される。

文字表示の位置座標

・ＬＯＣＡＴＥ文による文字表示位置の指定は文字の桁・行単位。画面の左上角を（０，０）とする。



・ＰＲＥＳＥＴ文による文字表示位置の指定はドット単位。画面の左上角を（０，０）とし、指定した位置を文字の表示パターンの左上角とする。



・ＰＵＴ　ＫＡＮＪＩ文は表示文字の指定と同時に位置の指定をすることができる。
位置の指定方法はＰＲＥＳＥＴ文と同じ。

漢字モードの文字表示

漢字モードは、文字表示の方法により、次の４種類に分かれています。

漢字モード	全角文字サイズ	半角文字サイズ	最大表示文字数	インターレース表示
ＫＡＮＪＩ０	16×16ドット	8×16ドット	32桁×13行	しない
ＫＡＮＪＩ１	12×16ドット	6×16ドット	40桁×13行	しない
ＫＡＮＪＩ２	16×16ドット	8×16ドット	32桁×24行	する
ＫＡＮＪＩ３	12×16ドット	6×16ドット	40桁×24行	する

※最大表示文字数は、漢字テキスト画面でＷＩＤＴＨ命令により最大桁数を指定し、ＫＥＹ　ＯＦＦ命令を実行させた状態での全角文字の表示数をあらわします。
半角文字の表示桁数は全角文字の２倍になります。
※インターレース表示は、一般の家庭用テレビではちらつきが大きく、実用的ではありません。

漢字モードでのＷＩＤＴＨ命令

・漢字テキスト画面でのみ有効です。漢字グラフィック画面で実行させるとエラーになります。
・指定する桁数は半角文字を基準にします。
・指定できる桁数の範囲は 26～ 64(または80)になります。
・漢字グラフィック画面では、同じ漢字モードの漢字テキスト画面で設定可能な最大の桁数・行数に自動的に設定されます。
・漢字テキスト画面の横方向の表示ドット数は、漢字モードとＷＩＤＴＨ命令によって変化します

ＫＡＮＪＩ０またはＫＡＮＪＩ２の場合

ＷＩＤＴＨ２６～３２のとき ……　256ドット
ＷＩＤＴＨ３３～６４のとき ……　512ドット

ＫＡＮＪＩ１またはＫＡＮＪＩ３の場合

ＷＩＤＴＨ２６～４０のとき ……　256ドット
ＷＩＤＴＨ４１～８０のとき ……　512ドット

漢字テキスト画面での文字の表示色

英数・かなモードでは、文字の表示色は前景色に固定されているため、ＣＯＬＯＲ命令で前景色の指定を変えると、テキスト画面に表示されているすべての文字の色が変わります。
漢字テキスト画面では、文字の表示色は「その文字を表示したときの前景色」になります。
前景色の指定を変えながら表示することにより、１文字ごとに表示色を変えることができます。

グラフィック画面

グラフィック画面の１ドット（画素）に対応するＶＲＡＭと表示色との関係は、次ページの図のようになっています。（図中、ーはＶＲＡＭ１バイトの各ビットをあらわします）

スクリーンNo.	VRAM1バイト	色　　数
2 / 4	(1ビット／1ドット)	8ドットを1単位とする領域ごとに1バイトのカラーテーブルを割り当て、色指定。(各領域ごとに16パレット中より2色の表示が可能) *1)
3	(4ビット／1ブロック)	各ブロックごとに16パレット
5	(4ビット／1ドット)	各ドットごとに16パレット
6	(2ビット／1ドット)	各ドットごとに4パレット
7	(4ビット／1ドット)	各ドットごとに16パレット
8	(8ビット／1ドット)	各ドットごとに 256色
10 *2) / 11 *2)	D1 Y1 A1 KL D2 Y2 A2 KH D3 Y3 A3 JL D4 Y4 A4 JH (4バイト／4ドット)	Aのビットが0の1バイトに対応する4ドットは、K,J(各6ビット)により4ドット単位で色指定し、Y(4ビット)により各ドットごとの輝度を指定 (YJK方式) Aのビットが1の1バイトに対応する1ドットは、Yのデータによる16パレット (RGB方式)
12	D1 Y1 KL D2 Y2 KH D3 Y3 JL D4 Y4 JH (4バイト／4ドット)	YJK方式 K、J(各6ビット)により、4ドット単位で色指定 Y(5ビット)により、各ドットごとに輝度を指定

*1) 8ドットを単位とする領域に対して3番目の色を指定すると、先に指定した色が変化します。

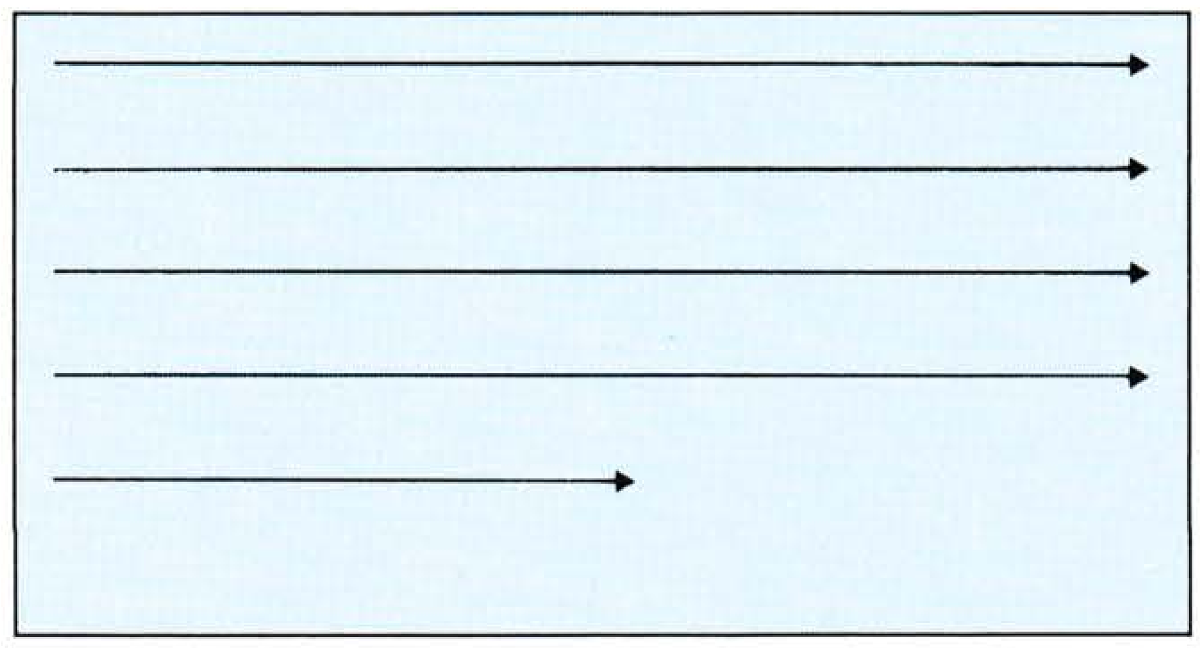
*2) スクリーン10と11の異なる点

スクリーン10：描画の際に指定されたカラーコードを、そのままパレットの番号とみなします。(カラーコードのデータは上位4ビット(Yの部分)にセットする)
各ドットに対応する1バイトのAは、自動的に1にします。

スクリーン11：ＢＡＳＩＣの描画命令（ＬＩＮＥやＰＡＩＮＴなど）を実行しても、Aは1に設定しません。したがって、描画の際に指定されたカラーコード（0～255）を、単なる1バイト分のデータとみなしてＹＪＫ方式で表示します。

参考　ＶＲＡＭの画面との対応

右の図のように画面の最上段の走査線の左端より右端、2段目の走査線の左端より右端‥‥の順で、各ドットのカラーコードが上の表にしめした形で格納されます。



色 指 定

表示の色は、**カラーコード**により指定します。あらかじめカラーコードに割り当てられている色を右の表に示します。（色の成分は、**R**（赤）、**G**（緑）、**B**（青）の各成分をそれぞれ0～7で示したものです。）

カラーコードと色との関係は固定されたものではなく、各カラーコードごとに512色中から1色を指定して色の割り当てを変えることができます。このような機能を**「パレット」**と呼びます。

色の割り当てを変更する命令がＣＯＬＯＲ＝文です。

例）カラーコード6（暗い赤）を赤にする。

ＣＯＬＯＲ＝（6，7，1，1）

赤　緑　青

カラーコード

色の成分

コード No.	最初の表示色	色の成分 赤	緑	青
0	透明	0	0	0
1	黒	0	0	0
2	緑	1	6	1
3	明るい緑	3	7	3
4	暗い青	1	1	7
5	明るい青	2	3	7
6	暗い赤	5	1	1
7	水色	2	6	7
8	赤	7	1	1
9	明るい赤	7	3	3
10	黄	6	6	1
11	明るい黄	6	6	4
12	暗い緑	1	4	1
13	紫	6	2	5
14	灰	5	5	5
15	白	7	7	7

◆カラーコード0の色を変更するときは、あらかじめ次の命令を実行させてください。

VDP(9)=VDP(9) OR &H20　　　（元に戻すとき）　VDP(9)=VDP(9) AND &HDF

ＶＲＡＭのページとＳＥＴ　ＰＡＧＥ命令

ＶＲＡＭは、ＳＣＲＥＥＮ5・6では4つの「ページ」に、ＳＣＲＥＥＮ7以上では2つの「ページ」に分かれています。グラフィック関係の命令は常に1つのページにだけ有効で、そのページを「描画ページ」と呼びます。また、実際に画面に表示されるのも常に1つのページだけで、そのページを「表示ページ」と呼びます。ＢＡＳＩＣがスタートしたときは、描画ページ・表示ページともにＶＲＡＭのページ0に設定されています。描画ページ・表示ページはＳＥＴ　ＰＡＧＥ命令で切り替えることができます。これにより、表示ページ以外のページを描画ページに設定して絵を描いておき、絵が完成してから表示ページを切り換えて、画面の表示が一瞬のうちに変わったように見せることができます。（⇨180ページ）

グラフィック関係の補足説明

漢字モードで使用できない命令

次の命令は、英数・かなモードでは使用できますが、漢字モードで実行させようとするとエラーになります。

ＣＬＳ …………… 漢字テキスト画面で実行させると、エラーになります。
漢字グラフィック画面では、エラーにならずに使用できます。
※漢字テキスト画面で画面表示を消去するときは、ＣＡＬＬ　ＣＬＳ命令を使用します。（⇨178ページ）

ＣＯＬＯＲ＝ …… ＣＯＬＯＲ文には、いくつかの機能がありますが、各カラーコードに割り当てられている色を変更する機能（パレット機能）は、漢字テキスト画面では使用できません。
例）　ＣＯＬＯＲ＝（８，１，６，１）などを実行するとエラーになる
漢字グラフィック画面では、支障なく使用できます。
その他の機能は、漢字テキスト画面でも使用できます。
※漢字テキスト画面でパレットの設定を変更したいときは、ＣＡＬＬ　ＰＡＬＥＴＴＥ命令を使用します。（⇨178ページ）

新たに追加された命令

グラフィック関係で新たに追加された命令で、本文中で説明していないものには、次の命令があります。

ＳＥＴ　ＳＣＲＯＬＬ（セット・スクロール） …… 画面をタテおよびヨコ方向にスクロールさせます。スクリーン５～８では、２ページ連続でヨコ方向にスクロールできます。

```
SET SCROLL 128,100,1,0
```

（ヨコのスクロール量）（タテのスクロール量）（目隠し指定）（次ページ指定）

ヨコのスクロール量は０～511の数値
タテのスクロール量は０～255の数値
目隠し指定は０か１。１の場合、水平スクロールしたとき左端の８ドットを周辺色で隠す。０の場合は隠さない。

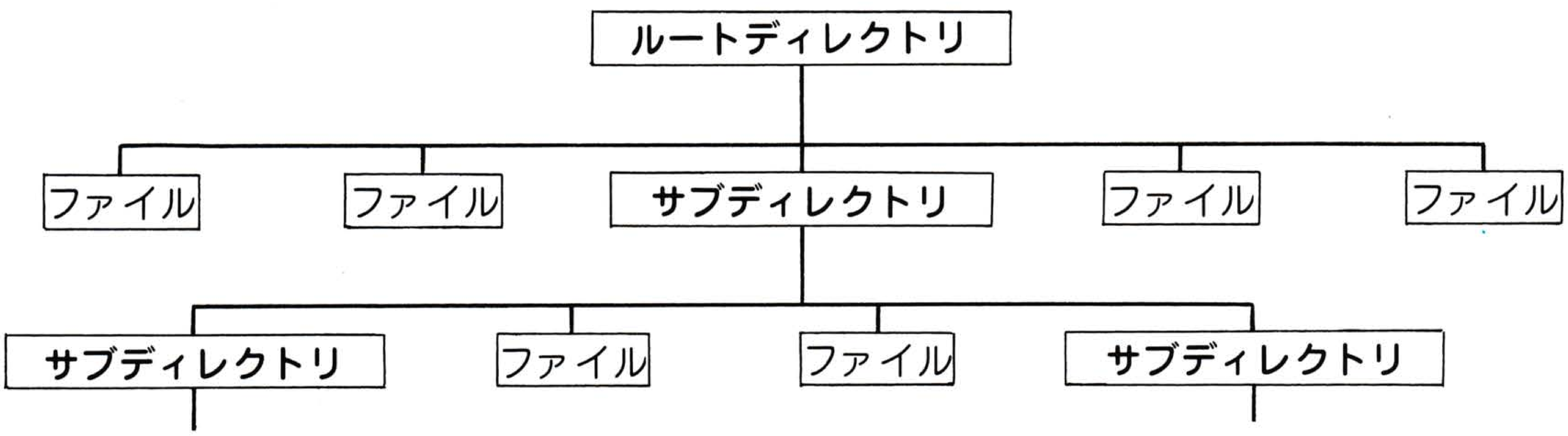
次ページ指定は０か１。０のときは水平スクロール時にそのページ内で画面の両端がつながってスクロールする。１のときは２ページ連続で水平スクロールする。ただし表示ページは奇数に設定する。

ファイル

パソコンに入力したプログラムやデータは、ファイルとしてフロッピーディスクに保存し､必要なときにパソコンのメモリの中に呼び出して再実行させることができます。MSX turbo Rでは、従来のMSX、MSX 2、MSX 2＋パソコンに比べ、ファイル管理機能が一段と強力になりました。

ディレクトリ

フロッピーディスクに保存されているファイルに関する情報は、それぞれのディスクの特定の領域にまとめて記録されています。FILES命令で表示されるのがこの情報です。この情報の集まりを**「ディレクトリ」**と呼び、MSX turbo Rのファイル管理の重要な基礎になっています。
人間やプログラムによる管理を容易にするため､MSX turbo Rでは､同じディスクに複数のディレクトリを作り、それぞれのディレクトリにファイルを分けて管理することができるようになっています。
フォーマット直後のディスクには基本となるディレクトリだけがあり、これを**「ルートディレクトリ」**と呼びます。
後から作られたディレクトリを**「サブディレクトリ」**と呼び、ルートディレクトリまたは他のサブディレクトリの下で、通常のファイルと同じように、ディレクトリ名などの情報を記録されます。
下の略図は、各ディレクトリやファイルの関係を示すものです。



ファイルの管理をディレクトリ別に行うため、あるディレクトリから他のディレクトリにあるファイルを直接に操作することはできません。その時、ファイルを直接操作できる状態にあるディレクトリを**「カレントディレクトリ」**と呼びます。BASICやDOSがスタートしたときには、カレントディレクトリは、ルートディレクトリになっています。
他のディレクトリにあるファイルを操作するには、次の2つの方法があります。
・カレントディレクトリを、目的のファイルがあるサブディレクトリに変更する。
・カレントディレクトリから目的のファイルがあるサブディレクトリまでの、すべてのディレクトリを経由して操作する。
　この場合、途中のサブディレクトリのディレクトリ名を正しい順番でファイル名の前に入力することが必要です。これを**「パスリスト」**と呼びます。パスリストはディレクトリ名を**「¥」**（エン）でつないで作ります。　**例）　RUN"LL¥MM¥NN¥FILE"**
　　　　　　　　　　　　　　　　　　　　　パスリスト　　ファイル名

ディレクトリ用の拡張命令

CALL　MKDIR（″＜ドライブ名＞＜パスリスト＞″）……新しいサブディレクトリの作成

例）CALL　MKDIR（″WORDS″）…………………新しいサブディレクトリ「WORDS」を作る

　　CALL　MKDIR（″WORDS¥TEXT″）…………「WORDS」の下に別のサブディレクトリ「TEXT」を作る

カレントディレクトリより下に新しいサブディレクトリを作る命令です。サブディレクトリを作ると、そのサブディレクトリの下に自動的に「.」と「..」というディレクトリ名のサブディレクトリが作られます。「.」はカレントディレクトリを、「..」は親ディレクトリ（すぐ上のディレクトリ）を表します。「.」と「..」は、パスリストの一部として利用することができます。

CALL　CHDIR（″＜ドライブ名＞＜パスリスト＞″）……カレントディレクトリを変更または表示する

例）CALL　CHDIR（″WORDS″）…………………カレントディレクトリを「WORDS」に変更する

　　CALL　CHDIR（″″）…………………………ルートディレクトリからカレントディレクトリまでのパスリストが表示される

　　CALL　CHDIR（″¥″）………………………ルートディレクトリに戻る

　　CALL　CHDIR（″..″）………………………すぐ上のディレクトリ（親ディレクトリ）に戻る

カレントディレクトリを別のディレクトリに変更する命令です。パスリストを省略すると、ルートディレクトリからカレントディレクトリまでのパスリストが表示されます。

CALL　RMDIR（″＜ドライブ名＞＜パスリスト＞″）……空のサブディレクトリを削除する

例）CALL　RMDIR（″TEXT″）…………………カレントディレクトリのすぐ下にある空のサブディレクトリ「TEXT」を削除する

「.」と「..」以外にファイルやサブディレクトリをもたないサブディレクトリを削除する命令です。ファイルやサブディレクトリをもつサブディレクトリを削除しようとするとエラーになります。ルートディレクトリは削除できません。

ファイルスペック

ファイルスペックはファイルの入出力の際に、対象となる入出力機器やファイルの名前を指定するためのものです。ファイルスペックは次のような形であらわします。

"デバイス名：
パスリスト￥ファイル名．拡張子"

ファイルスペックは必ず「"」で囲みます。
デバイス名とパスリストのあいだは「：」（コロン）で、パスリストとファイル名の間は「￥」で、ファイル名と拡張子の間は「．」（ピリオド）で区切ります。

デバイス名・パスリスト・ファイル名・拡張子は、それぞれ省略できる場合があります。

	デバイス名 ＊1	ファイル名 ＊1	拡張子 （⇨38ページ）
フロッピーディスク	Ａ：～Ｇ：	8文字以内 ・英字の大文字、小文字を区別しない 例）"adc"と"ABC"は同じファイル名	3文字以内 ＊2
ＲＡＭディスク	Ｈ：		
互換ＲＡＭディスク	ＭＥＭ：		
テキスト画面	ＣＲＴ：	6文字以内 （省略可能）	なし （使用できない）
グラフィック画面	ＧＲＰ：		
プリンタ画面	ＬＰＴ：		

＊1）デバイス名を省略した場合 ……ドライブＡ（Ａ：）になる
＊2）ファイル名に使用できる文字 ……英字、数字、カタカナ（漢字モードのとき：漢字、ひらがな）
＊3）パスリストが必要な場合 ………操作するファイルがカレントディレクトリにないとき

メモ： ＲＡＭディスク「Ｈ：」は、「ＣＡＬＬ　ＲＡＭＤＩＳＫ」命令（⇨171ページ）によりＲＡＭディスクを設定してから使用してください。

ドライブ名

入出力機器にフロッピーディスクドライブを使用するときのデバイス名を、特に**「ドライブ名」**と呼びます。ドライブ名は、使用するドライブの接続のしかたにより異なります。

- **ドライブ１台が内蔵または接続されているとき** …… ドライブ名はＡ：になります。ただし、ＣＯＰＹ文などではＢ：として使うことができます。
- **ドライブ２台が内蔵されているとき** ……………… 片方がＡ：、他方がＢ：になります。
- **ドライブ１台が内蔵されているパソコンのスロットにドライブをもう１台接続したとき** ……… スロットに接続したドライブがＡ：、内蔵ドライブがＣ：になります。

拡張子について

拡張子は、ファイルの種類を区別するための分類記号です。文字数と使用可能な文字種以外、特に決められていません。使用するかどうか、どんな拡張子にするかは自由です。
たとえば下のように使い分けると、ファイルの数が多くなったときに便利です。

例)

ＢＡＳＩＣプログラム	……	ＢＡＳ
機械語プログラム	…………	ＢＩＮ
データファイル	…………	ＤＡＴ

拡張子は省略できますが、拡張子をつけて保存したファイルを呼び出すときには省略できません。

ご注意：「ＢＡＴ」「ＳＹＳ」「ＣＯＭ」は、ＭＳＸ－ＤＯＳ２で特別な意味を持つ拡張子ですので使用しないでください。

ワイルドカード

ファイルを操作する命令（ＦＩＬＥＳやＣＯＰＹなど）では、ファイル名や拡張子に**「＊」**や**「？」**を使用することにより、複数のファイルを指定することができます。

＊（アスタリスク）…… 文字数の決まっていない不特定の文字列を表します。したがって「＊．＊」は、「そのディレクトリ内のすべてのファイル」を指定したことになります。

？（クエスチョン）…… 不特定の１文字を表します。番号だけ異なるファイルなどの指定に使います。

このような機能を**「ワイルドカード」**と呼びます。

例)

FILES "ABC.*"	……………	ABC.BAS ABC.BIN
FILES "?BC.BAS"	…………	ABC.BAS DBC.BAS
COPY "A:*.*" TO "B:"	……	カレントディレクトリ内の全ファイルをコピーする

資料編

保存と呼び出しの命令

ＢＡＳＩＣプログラムの保存・呼び出しの命令

・保存　　ＳＡＶＥ
・呼び出し　ＬＯＡＤ，ＲＵＮ，ＭＥＲＧＥ（アスキー型式のファイル専用）

機械語プログラムやバイナリーデータ、ＶＲＡＭデータの保存・呼び出しの命令

・保存　　ＢＳＡＶＥ，ＣＯＰＹ（ＶＲＡＭデータまたは配列変数領域とファイル間）
・呼び出し　ＢＬＯＡＤ，ＣＯＰＹ（ファイルとＶＲＡＭデータまたは配列変数領域間）

数値や文字列データの保存と呼び出しの命令

・保存　　ＰＲＩＮＴ＃，ＰＲＩＮＴ＃　ＵＳＩＮＧ
・呼び出し　ＩＮＰＵＴ＃，ＩＮＰＵＴ＄，ＬＩＮＥ　ＩＮＰＵＴ＃

グラフィック画面やテキスト画面、プリンタへの文字の出力（出力専用）

ＰＲＩＮＴ＃，ＰＲＩＮＴ＃　ＵＳＩＮＧ

シーケンシャルファイルとランダムファイル

データを保存するファイルには、シーケンシャルファイルとランダムアクセスファイルの２種類があり、どちらも一定の手順で操作することが必要です。

・シーケンシャルファイル

データを常に決まった順番で保存・呼び出しする方式のファイルです。ファイルの長さを自由に変えることができます（可変長のファイル型式）が、保存したデータを**一部分だけ変更したり削除することはできません。**

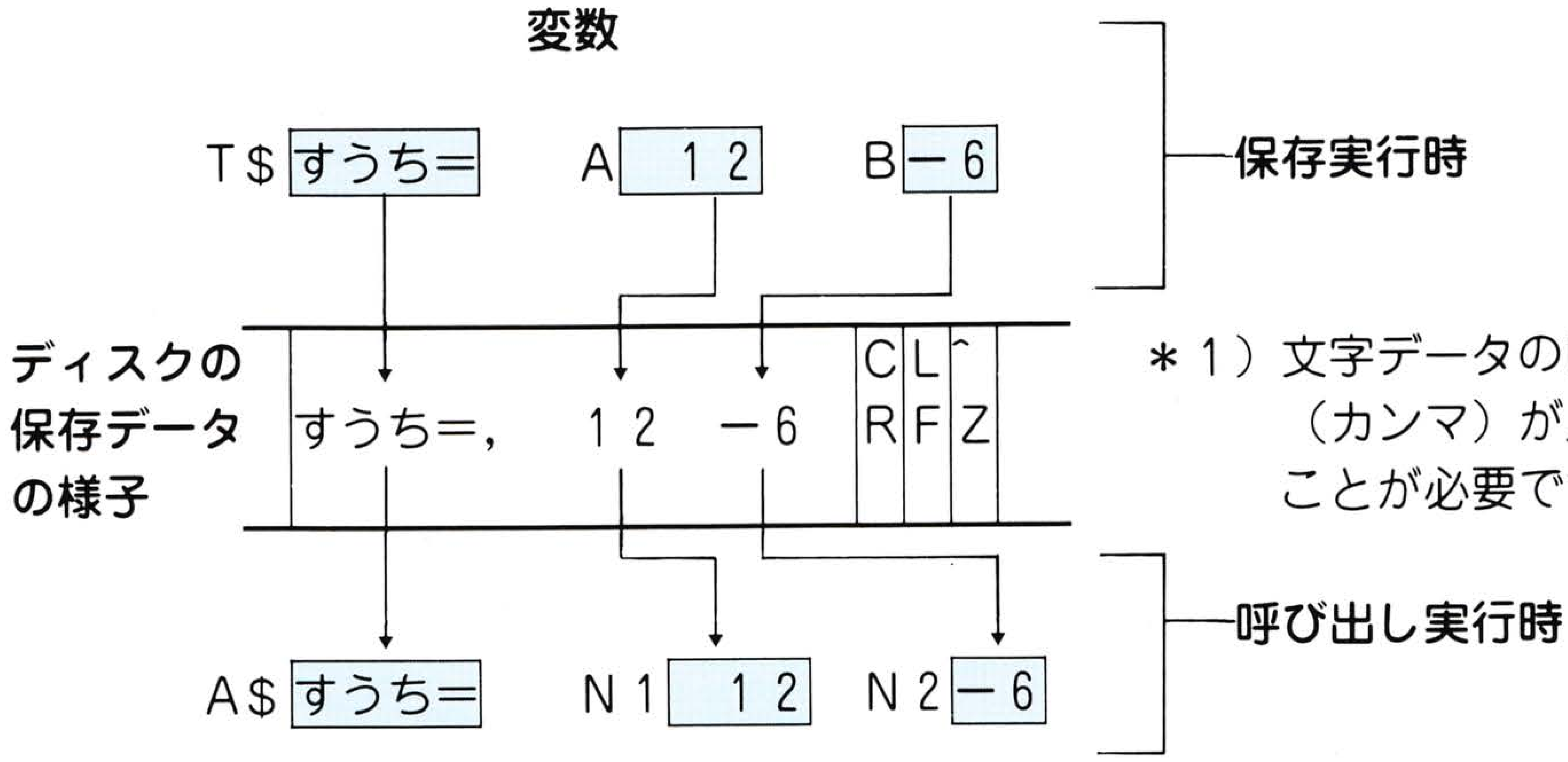
プログラム例）

保存

```
100 OPEN "A:FILE.DAT" FOR OUT
PUT AS #1
110 PRINT #1, T$;", ";A;B
120 CLOSE #1
```

呼び出し

```
100 OPEN "A:FILE.DAT" FOR INP
UT AS #1
110 INPUT #1, A$, N1, N2
120 CLOSE #1
```



＊1）文字データの区切りには、必ず「，」（カンマ）が入力されるようにすることが必要です。

＊2）複数のデータを保存・呼び出しする場合、ＯＰＥＮ，ＣＬＯＳＥを各データごとに実行する必要はありません。同じファイル番号を使う操作の最初と最後に一度ずつ実行するだけでかまいません。
ただし、操作中にディスクを入れ替えたり、ＥＮＤ命令以外でプログラムを終了する場合は事前にＣＬＯＳＥ命令を実行するようにしてください。これらの操作を、ファイルを開いたままで行うと、ファイルのデータが壊れることがあります。

・ランダムアクセスファイル

ファイルを一定の長さの部分（**「レコード」**）に区切って番号で管理することにより、保存や呼び出しを必要な部分から行えるようにしたファイルです。何らかのキー項目にしたがって保存位置を決定することにより、呼び出し時に必要なデータをすぐに取り出すことができます。
固定長のファイル型式になります。また、この型式のファイルは、**文字型のデータしかあつかうことができない**ので、数値型データを保存する場合は、あらかじめ文字型データに変換してから保存します。呼び出し時には、逆に文字型データから数値型データに変換してからデータを利用します。この方式のファイルは、**フロッピーディスクでだけ使用できます**。

（プログラム例）

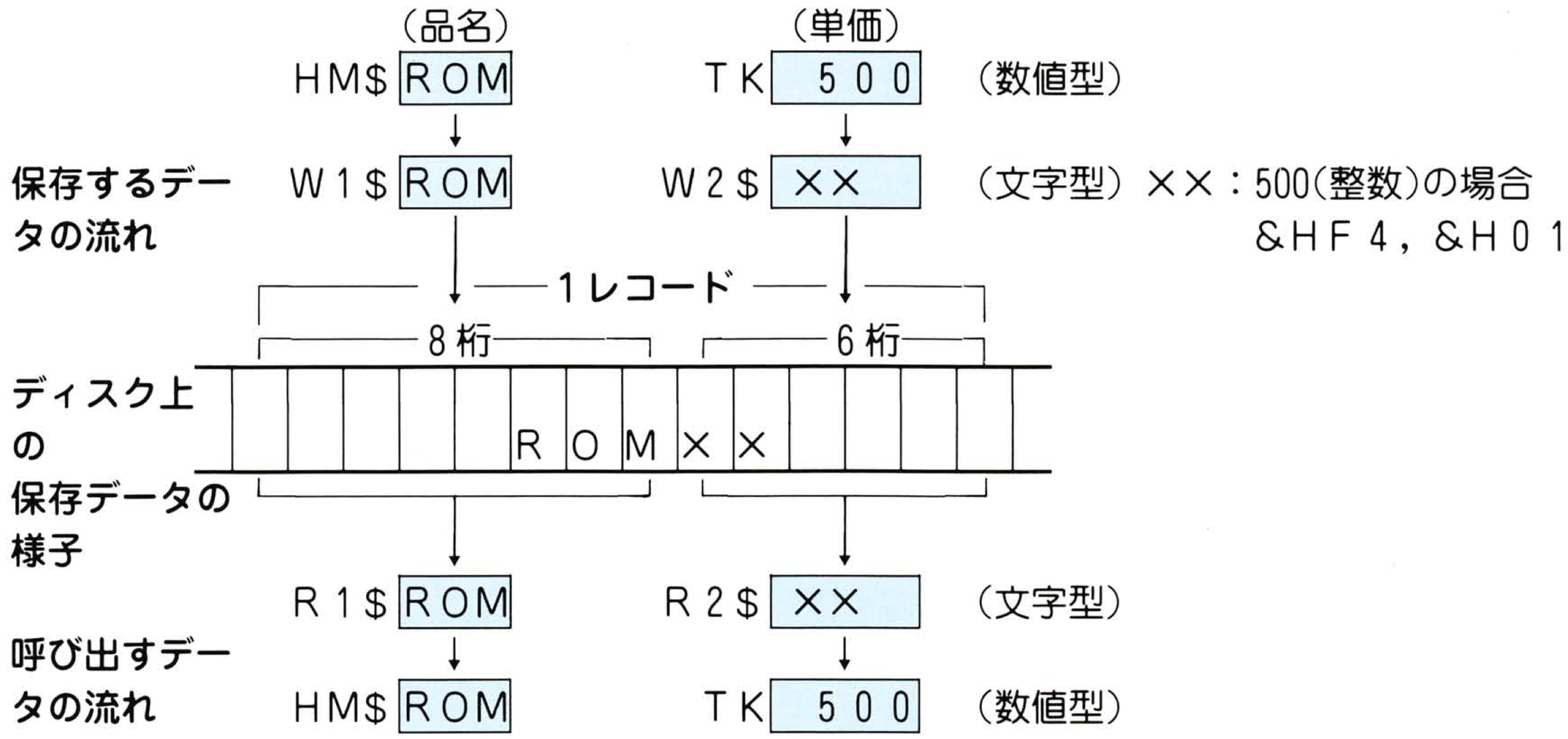
保存

```
100 OPEN "A:FILE.DAT" AS #1
110 FIELD #1, 8 AS W1$, 6 AS W2$
120 RSET W1$=HM$
130 LSET W2$=MKI$(TK)              *1)
140 PUT #1, R                      *2)
150 CLOSE #1
```

呼び出し

```
100 OPEN "A:FILE.DAT" AS #1
110 FIELD #1, 8 AS R1$, 6 AS R2$
120 GET #1, R
130 HM$=R1$
140 TK=CVI(R2$)                    *3)
150 CLOSE #1
```

資料編



＊1）変数の型により、MKI$，MKS$，MKD$関数のいずれかを使用します。
＊2）変数Rは、保存・呼び出しするデータの部分の番号（＝レコード番号：1以上の正の整数）として使っています。
＊3）変数の型により、CVI，CVS，CVD関数のいずれかを使用します。

・複数のデータを保存・呼び出しする場合、OPEN，CLOSEを各データごとに実行する必要はありません。同じファイル番号を使う操作の最初と最後に一度ずつ実行するだけでかまいません。
・複数のファイルを同時に開く場合（＃1と＃2を同時に使用するときなど）には、事前にMAXFILES文で同時に使用するファイル数の宣言が必要です。（**例** MAXFILES=2）
MAXFILES文は、なるべくプログラムの先頭で実行させてください。プログラムの途中で実行させると、変数の値が失われる場合があります。
・FILED文で割り当てることができるデータの長さの合計は、256バイトまたはOPEN文で指定したレコード長（256バイト以下）までです。

RAMディスク

ＲＡＭディスク（ラムディスク）は、メモリのうちのＢＡＳＩＣで使用しない部分をプログラムやデータの保存領域として、フロッピーディスクのように使用する機能です。保存や呼び出しの操作のしかたは、フロッピーディスクとほぼ同じです。（デバイス名はＨ：⇨166ページ）
パソコンの電源スイッチを切ると、ＲＡＭディスクに保存したプログラムやデータは消えてしまいますが、保存や呼び出しの速度が非常に早いので、うまく使うと便利です。

ＲＡＭディスク用の拡張命令

ＣＡＬＬ　ＲＡＭＤＩＳＫ（<容量>） ……………………………ＲＡＭディスクを作る

例）ＣＡＬＬ　ＲＡＭＤＩＳＫ（９６）　……………………………９６キロバイトの大きさのＲＡＭディスクを作る

メモリ内にＲＡＭディスク用の領域を設定する命令です。ＲＡＭディスクを使用する場合は、パソコンの電源を入れてからＲＡＭディスクを使い始める前に、必ずこの命令を実行してください。
設定したＲＡＭディスクには、ＬＯＡＤ，ＳＡＶＥ，ＲＵＮ，ＦＩＬＥＳなどのフロッピーディスクに対する命令を、ドライブ名を「Ｈ：」にすることにより使用できます。

ＲＡＭディスクに保存した内容は、パソコンの電源スイッチを切るか、リセットボタンを押したときに消えてしまいます。ＢＡＳＩＣを終了する前に、ＲＡＭディスクの内容をフロッピーディスクにコピーしておいてください。
例）ＣＯＰＹ　″Ｈ：＊．＊″　ＴＯ　″Ａ：″

ＲＡＭディスクの大きさは、１６キロバイト単位に端数を切り上げて設定されます。設定できる最大の大きさはパソコンの機種により異なりますが、限度を越える設定をしても最大の大きさに設定されます。

ご注意：ＲＡＭディスクを作ってから、ＣＡＬＬ　ＲＡＭＤＩＳＫ命令によりＲＡＭディスクの大きさを変更することはできません。「ＣＡＬＬ　ＲＡＭＤＩＳＫ（０)」と入力して、ＲＡＭディスク用の領域を消去してから、ＲＡＭディスクを作り直してください。（このとき、ＲＡＭディスクに保存されていたファイルは消えてしまいます。）

サブルーチンと割り込み

多くのプログラムは、いくつかの決まった手順の作業（処理）の組み合わせになります。プログラムの中である処理を繰り返して行う場合は、その処理を一か所に置いて必要が生じるたびにその部分にジャンプし、処理が終わったら元の部分に戻るようにすると、簡潔で理解しやすいプログラムが作れます。このようにした処理のことを**「サブルーチン」**と呼びます。
また、プログラムを実行中に特別なできごとが起きたとき、実行中の処理をただちに中断して別の処理を行い、その処理が終わったら元の処理の実行を再開することが必要な場合があります。
これを**「割り込み」**と呼びます。

サブルーチン

サブルーチンを使用するための命令として、次のものがあります。

GOSUB　行番号 ········ ＧＯＳＵＢ（ゴー・サブ）は、**行番号**でしめす行にジャンプし、その行から続く命令をＲＥＴＵＲＮ文にあうまで実行します。サブルーチンへジャンプするのに用います。
行番号は省略できません。存在しない行番号は指定できません。

RETURN　行番号 ······ ＲＥＴＵＲＮ（リターン）は、必ずＧＯＳＵＢ文と組み合わせて用いる命令です。サブルーチンの実行を終了するときに使用します。**行番号**は省略できます。省略したときはＧＯＳＵＢ文の次の文に戻って実行します。行番号を指定したときはその行番号にジャンプします。

ON　変数または計算式　GOSUB　行番号，行番号，行番号····
ＯＮ　ＧＯＳＵＢ（オン～ゴー・サブ）は、変数の値によって異なるサブルーチンにジャンプする命令です。行番号の並びは左側から順番に１から始まる数に対応します。つまり、変数の値が３なら３番目の行番号にジャンプします。値が０または行番号の個数より大きいときは、ジャンプせずに次の行に進みます。値がマイナスのときはエラーになります。サブルーチンの実行を終了するときはＲＥＴＵＲＮ文を使用します。

・実行されるＧＯＳＵＢ文とＲＥＴＵＲＮ文の数が一致するように注意してください。対応するＧＯＳＵＢ文なしにＲＥＴＵＲＮ文を実行させるとエラーになります。
・サブルーチンから他のサブルーチンにジャンプすることもできます。

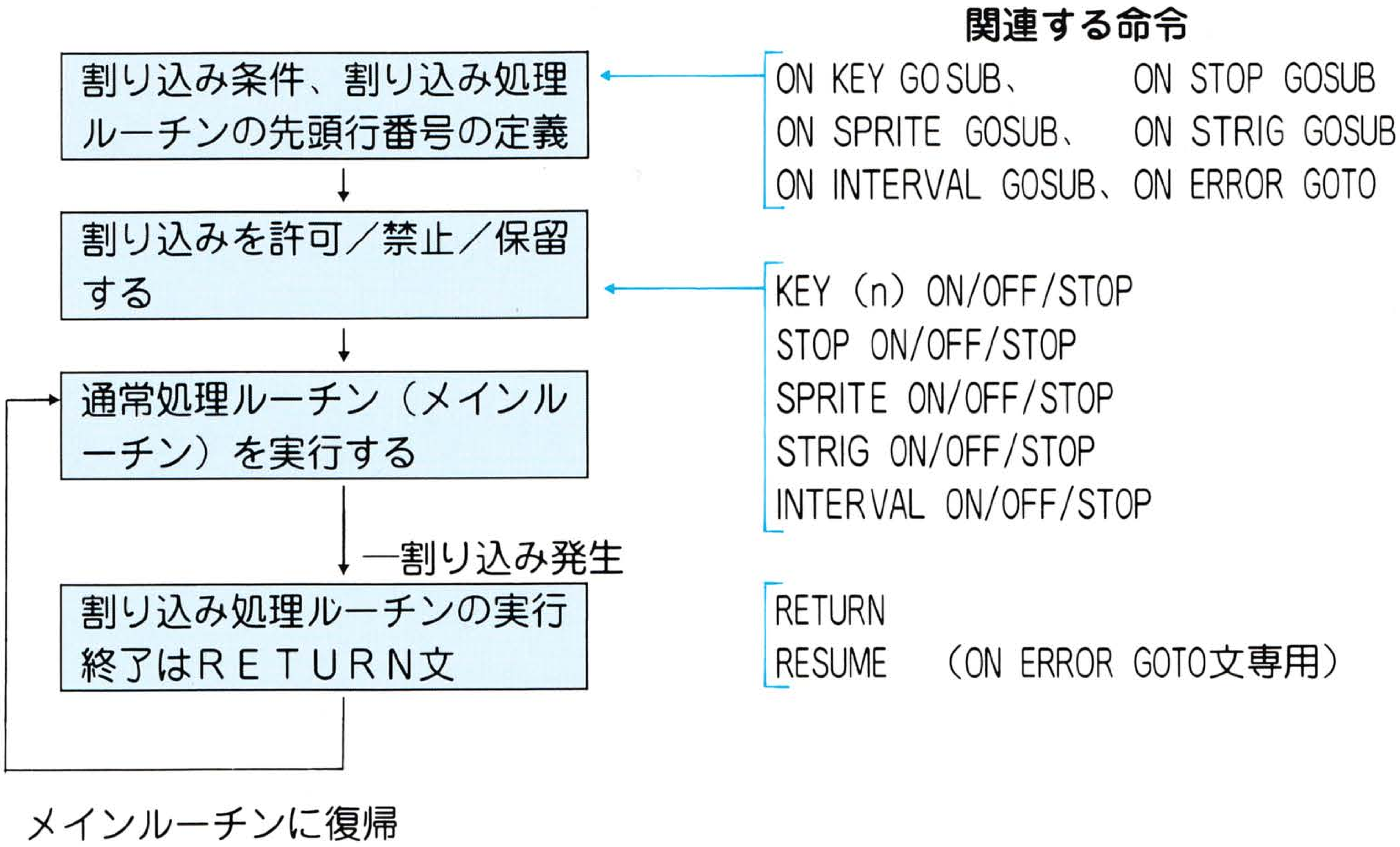
割り込み

通常、あることが起きたときに特定の処理を実行するときは、そのできごとが起きそうな部分でＩＮＫＥＹ$関数などを使って実際に起きているかどうかを調べます。しかし、これでは他の処理を実行中にそのできごとが起きたときには対処できません。そこでＢＡＳＩＣでは、一定のできごとが起きたとき、すぐに実行中の処理を中断して指定されたサブルーチンにジャンプするように設定する命令がいくつか用意されています。このような処理を**「割り込み」**と呼びます。

たとえばファンクションキーが押された時に、対応するサブルーチンにジャンプするようにあらかじめ設定しておくと、いつファンクションキーが押されてもその時点で割り込みがかかり、現在実行中の処理を中断して指定されたサブルーチン（割り込み処理ルーチン）にジャンプして新しい処理（たとえばミサイルの発射など）を実行したあと、中断していた処理を再開します。

ＭＳＸのＢＡＳＩＣで使用できる割り込みには、**ファンクションキー割り込み**、**ストップキー割り込み**、**スプライト衝突割り込み**、ジョイスティックなどの**トリガー割り込み**、**インターバルタイマー割り込み**があります。

割り込み使用時の処理の流れを下の図でしめします。



資料編

キャラクタコード表

上位 4 ビット（16進数）→

下位 4 ビット（16進数）↓

	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0	@	P	`	p	♠			ー	タ	ミ	た	み
1	!	1	A	Q	a	q	♥	あ	。	ア	チ	ム	ち	む
2	"	2	B	R	b	r	♣	い	「	イ	ツ	メ	つ	め
3	#	3	C	S	c	s	♦	う	」	ウ	テ	モ	て	も
4	$	4	D	T	d	t	○	え	、	エ	ト	ヤ	と	や
5	%	5	E	U	e	u	●	お	・	オ	ナ	ユ	な	ゆ
6	&	6	F	V	f	v	を	か	ヲ	カ	ニ	ヨ	に	よ
7	'	7	G	W	g	w	ぁ	き	ァ	キ	ヌ	ラ	ぬ	ら
8	(	8	H	X	h	x	ぃ	く	ィ	ク	ネ	リ	ね	り
9	)	9	I	Y	i	y	ぅ	け	ゥ	ケ	ノ	ル	の	る
A	*	:	J	Z	j	z	ぇ	こ	ェ	コ	ハ	レ	は	れ
B	+	;	K	[	k	{	ぉ	さ	ォ	サ	ヒ	ロ	ひ	ろ
C	,	<	L	¥	l	¦	ゃ	し	ャ	シ	フ	ワ	ふ	わ
D	−	=	M	]	m	}	ゅ	す	ュ	ス	ヘ	ン	へ	ん
E	.	>	N	^	n	~	ょ	せ	ョ	セ	ホ	゛	ほ	
F	/	?	O	_	o		っ	そ	ッ	ソ	マ	゜	ま	■

←カーソル（&HFF の ■）

グラフィックキャラクタ

	4	5
0		π
1	月	┴
2	火	┬
3	水	┤
4	木	├
5	金	┼
6	土	│
7	日	─
8	年	┌
9	円	┐
A	時	└
B	分	┘
C	秒	╳
D	百	大
E	千	中
F	万	小

〈参考〉
特殊キーの
キーコード

キー		コード
BS		&H8
TAB		&H9
HOME		&HB
RETURN		&HD
INS		&H12
SELECT		&H18
ESC		&H1B
カーソルキー	→	&H1C
	←	&H1D
	↑	&H1E
	↓	&H1F
DEL		&H7F

例）「A」のキャラクタコードは、上位 4 ビットが「4」、下位 4 ビットが「1」なので、
&H41（10進数では、16× 4 ＋ 1 ＝65）
「¥」のキャラクタコードは、上位 4 ビットが「5」、下位 4 ビットが「C」なので、
&H5C（10進数では、16× 5 ＋12＝92：16進数の A～F は10進数の10～15に対応）

グラフィックキャラクタの入出力

グラフィックキャラクタは、キャラクタコードの前にグラフィックヘッダ（&H01）を付けて入出力されます。

たとえば、「時」のキャラクタコードは&H4Aですから、

キーボードからの入力　　：&H01、&H4Aの 2 バイトが入力されます。
テレビ、プリンタへの出力：&H01、&H4Aの 2 バイトが出力されます。

例） テキスト画面へ「時」を表示　　PRINT CHR$(1);CHR$(&H4A)

ご注意： 漢字モードでは、グラフィックキャラクタは使用できません。

BASICリファレンス表

以下のページでは、漢字ＢＡＳＩＣの命令や関数などを、用途別に分類して書式（入力の形式）、機能、文例（実際の入力例）を表にまとめています。

リファレンス表の見かた

必要な機器のマーク

特定の周辺機器を接続または内蔵していないと使用できない命令や関数については、以下のように必要な機器のマークを表示しています。

> ：フロッピーディスク　　：プリンタ　　漢：漢字プリンタ

書　式

命令を入力するときの書きかたを示します。実際の入力時には、次のような決まりにしたがってください。

- **英字の大文字**で示された項目は、そのとおりに入力します。大文字でも小文字でもかまいません。
- **カギカッコ＜　＞**で示された項目は、ユーザーが指定する数値または文字列をあらわします。
- **角カッコ［　］**で示された項目は、省略することができます。省略したときは、初期値（ＢＡＳＩＣが設定する値）または最後に設定した値が適用されます。
- **棒線｜　｜**で囲まれた項目は、｜　｜内の項目から１つを選んで入力します。
- 下のように**「，」（カンマ）**で区切られている項目を省略するときは、省略する項目の位置によって省略のしかたが変わります。

  ＣＯＬＯＲ　［＜前景色＞］［，＜背景色＞］［，＜周辺色＞］

  以後の項目をすべて省略するときは、「，」を含めて省略します。
  後ろの項目を省略しないで入力するときは、それ以前の「，」を省略しないで入力します。

  例）　ＣＯＬＯＲ　，，７

- **「;」（セミコロン）**で区切られている項目を省略するときは、「;」も含めて省略します。
- **「・・・」**は同じ項目の連続の省略をあらわしています。１行の許す長さ（255文字）の範囲内で任意の回数だけ繰り返すことができます。
- 上記のカギカッコ、角カッコ、棒線、連続記号は入力してはいけません。それ以外の記号は、すべてそのまま入力します。

機　能　命令の機能を簡単に示します。

文　例　実際の入力のしかたの見本として基本的な入力例を示します。

命　令

プログラム作成

書　式	機　能	文　例
AUTO　[〈先頭の行番号〉][,〈増分値〉]	行番号を自動的に発生する	AUTO　100, 20
DELETE　[〈始めの行番号〉] ー[〈終わりの行番号〉]	プログラムの一部を行単位で削除する	DELETE　50ー100
KEY　LIST	ファンクションキーの内容を表示する	KEY　LIST
LIST　[〈先頭の行番号〉] ー[〈最終の行番号〉]	メモリ上にあるプログラムのリストを表示する	LIST　100ー200
LLIST　[〈先頭の行番号〉] ー[〈最終の行番号〉]	メモリ上にあるプログラムのリストをプリンタで印刷する	LLIST
NEW	メモリ上のプログラムを消す	NEW
REM　[〈注釈文〉]　（「′」で代用可能）	プログラムに注釈を入れる	REM　＊　これはサンプルです　＊
RENUM [〈新先頭行番号〉] [,〈旧先頭行番号〉][,〈増分値〉]	プログラムの行番号を付け直す	RENUM　100, 10, 5

定義および設定

（「互換RAMディスク」とは、従来のMSX2、MSX2＋BASICと互換性のあるRAMディスクのことです。）

書　式	機　能	文　例
CALL　ANK	英数・かなモードに設定する	CALL　ANK
CALL　KANJI[〈漢字モード〉]	漢字モードに設定する（モードは0～3）	CALL　KANJI
CALL　MEMINI [〈番地〉]	互換RAMディスクの使用を宣言	CALL　MEMINI
CALL　RAMDISK (〈容量〉)	RAMディスクの大きさを設定する	CALL　RAMDISK (128)
CLEAR　[〈文字列領域の大きさ〉 [,〈メモリの上限〉]]	変数の初期化およびメモリ領域の大きさを設定する	CLEAR　300, &HE000
DEF　FN〈名前〉[(〈引数〉)] [,〈引数〉…])]=〈関数の定義式〉	関数を定義する	DEF　FNA (X, Y) =X＊2/Y
DEF INT / SNG / DBL / STR　〈文字〉[ー〈文字〉] [,〈文字〉[ー〈文字〉]] …	頭文字で指定した変数の変数型を定義する （INT:整数型、SNG:単精度型 DBL:倍精度型、STR:文字型）	DEFINT　AーZ
DEF　USR[〈番号〉] =〈開始アドレス〉	ユーザー作成の機械語サブルーチンの開始アドレスを定義	DEF　USR1=&HF000
DIM〈配列変数名〉(〈添字の最大値〉 [,〈添字の最大値〉…])	配列の名前、型、大きさ、次元を定義する	DIM　A (10, 20)
ERASE　〈配列変数名〉 [,〈配列変数名〉…]	配列変数を消去し、割り当てたメモリを解放する	ERASE　A, B, C
KEY〈キー番号〉,〈文字列〉	ファンクションキーに文字列を定義する	KEY1, "FILES"+ CHR$ (13)

データ入出力（ファイル関係を除く）

書　式	機　能	文　例
DATA〈定数〉[,〈定数〉…]	READ文で読みとる数値や文字列を与える	DATA　123, ABC, キミ

書　式	機　能	文　例
INPUT [″〈プロンプト文〉″;]〈変数名〉 [,〈変数名〉…]	キーボードからデータを入力し、指定した変数に入れる	INPUT　″すうりょうは″;N
LINE INPUT [″〈プロンプト文〉″;] 〈文字型変数名〉	キーボードから文字列を入力し、文字型変数に入れる	LINE　INPUT　″ナマエ″;A$
LPRINT [〈式〉[,〈式〉]…]	プリンタで印刷する	LPRINT　A，B$;C
MID$（〈文字型変数〉,〈式1〉,[〈式2〉]）＝ 〈文字列〉	文字型変数の〈式1〉番目の文字から〈式2〉個の文字を〈文字列〉で置き換える	MID$（A$，2，3）＝″BCD″
READ〈変数名〉[,〈変数名〉…]	DATA文からデータを読み取り、変数に入れる	READ　A，B$，C$
RESTORE [〈行番号〉]	次にREAD文で読み取るデータを指定する	RESTORE　1000

プログラムの実行・流れのコントロール

書　式	機　能	文　例
CALL　PAUSE（〈時間〉）	指定された時間（1/1000秒単位）プログラムの実行を停止	CALL　PAUSE（5000）
CONT	CTRL ＋ STOP またはSTOP文により停止させたプログラムの実行を再開	CONT
END	プログラムの実行を終了する OPENしたままのファイルがあれば、CLOSEする	END
FOR〈変数名〉＝〈開始値〉 TO〈終了値〉[STEP〈増分値〉] NEXT [[〈変数名〉][,〈変数名〉]…]	FOR文からNEXT文までの一連の命令を、指定した回数だけ繰り返し実行	FOR A＝0 TO 15 STEP 2 ： NEXT A
GOSUB〈行番号〉	サブルーチンにジャンプ	GOSUB　100
GOTO〈行番号〉	指定された行番号にジャンプ	GOTO　100
IF〈条件〉THEN |〈文〉/〈行番号〉| [ELSE |〈文〉/〈行番号〉|]	条件判断を行う 〈条件〉が真(-1)ならTHEN文またはGOTO文を実行 〈条件〉が偽(0)ならELSE文を実行 ELSE文省略時は次の行に進む	IF A$＝″Y″ THEN 200 ELSE 300 （A$が″Y″なら200行へ A$が″Y″以外なら300行へ）
IF〈条件〉GOTO |〈行番号〉| [ELSE |〈文〉/〈行番号〉|]		
ON〈式〉GOSUB 〈行番号〉[,〈行番号〉…]	〈式〉の値によって異なるサブルーチンにジャンプする	ON　A　GOSUB 100，200，300
ON〈式〉GOTO 〈行番号〉[,〈行番号〉…]	〈式〉の値によって異なる行番号にジャンプする	ON　A　GOTO 100，200，300
RETURN [〈行番号〉]	サブルーチンから復帰する	RETURN
RUN [〈行番号〉]	メモリ内のプログラムを実行 行番号指定時はその行から	RUN
RUN ″〈ファイルスペック〉″	デバイス名で指定された機器からファイル名で指定されたプログラムをロードして実行	RUN ″A：PROG1.BAS″
STOP	プログラムの実行を停止してダイレクトモードに戻る	STOP
TROFF	TRONを解除する	TROFF
TRON	実行した行番号を画面に表示	TRON

表示関係

(t)はテキスト画面専用。(g)はグラフィック画面専用。(e)は漢字テキスト画面では使えない。

書式	機能	文例
CIRCLE [STEP] (x, y),〈半径〉 [,〈カラーコード〉] [,〈開始角度〉] [,〈終了角度〉] [,〈比率〉] (g)	円またはだ円、扇形を描く (x, y) は中心座標	CIRCLE (80, 80), 40, 2
CALL CLS	画面を消去する	CALL CLS
CALL PALETTE [(〈カラーコード〉,〈R〉,〈G〉,〈B〉)]	カラーコードの色を変える 〈R〉〈G〉〈B〉は0～7の数値 項目を全部省略したときは、色を初期状態に戻す	CALL PALETTE (7, 7, 0, 0) (カラーコード7を赤に)
CLS (e)	画面を消去する	CLS
COLOR [〈前景色〉] [,〈背景色〉] [,〈周辺色〉]	画面の各部の色を設定する (初期値は、15,4,7)	COLOR 15, 2, 10
COLOR =(〈カラーコード〉,〈R〉,〈G〉,〈B〉) (e)	カラーコードの色を変える 〈R〉〈G〉〈B〉は0～7の数値	COLOR=(8, 0, 7, 0) (カラーコード8を緑色に)
COLOR [=NEW] (e)	カラーコードの色を初期状態に戻す	COLOR
COLOR SPRITE (〈スプライト面番号〉)=〈カラーコード〉 (g)	指定したスプライト面のスプライトパターンの色を変える	COLOR SPRITE (0)=10
COLOR SPRITE$ (〈スプライト面番号〉)=〈カラーコード列〉 (g)	指定したスプライト面のスプライトパターンの1行毎の色を変える	COLOR SPRITE$ (0) =CHR$ (10) +CHR$ (3) … (スプライトパターンの1行目をカラーコード10に、2行目をカラーコード3に)
DRAW〈文字列〉 (g)	画面上の参照点を相対的に移動させながら線を描く U,D,L,R,E,F,G,H,M,B,N,A,C,S,X;などのマクロ命令と数字からなる文字列で指定する	DRAW "C2U9L25D9R25"
KEY ON KEY OFF (t)	ファンクションキーの内容を画面最下行に表示するかしないかを指定する	KEY ON (表示する) KEY OFF (表示しない)
LINE [STEP] [(x1, y1)] −(x2, y2) [,〈カラーコード〉] [, {B / BF}] [,〈論理演算子〉] (g)	直線または四角形を描く Bをつけると2点を対角線とする四角形を描く BFをつけると四角形の内側を指定された色で塗りつぶす	LINE (0, 0) − (100, 100) , 2, B
LOCATE [〈桁〉] [,〈行〉] [,〈カーソルスイッチ〉] (t)	テキスト画面のカーソルの位置を指定する	LOCATE 14, 10
PAINT [STEP] (x, y) [,〈領域色〉] [,〈境界色〉] (g)	指定された〈境界色〉で囲まれた領域を、指定された〈領域色〉で塗りつぶす	PAINT (50, 50), 8, 2 (境界色で囲まれていない領域を指定すると、画面全体を塗りつぶします)
PRESET [STEP] (x, y) [,〈カラーコード〉] [,〈論理演算子〉] (g)	指定された位置に描かれた点を消す	PRESET (100, 100)
PRINT [〈式または文字列〉] [{, / ;} 〈式または文字列〉…] (t)	文字列や数値をテキスト画面に表示する	PRINT "ABC", 123;"4"

書　　式	機　能	文　例
PRINT USING〈書式〉; 〈式または文字列〉 [|,|〈式または文字列〉…] |;| (t) 書式制御文字 ″!″　:最初の1字を表示 ″&<n個の空白>&″ :n+2字分の文字を表示 ″@″　:文字列全部を表示 ″#″　:数値の表示桁数を指定 ″ ″　:小数点の位置指定桁ぞろえに使用 ″+″　:数値の前(後)に+,-をつけて表示 ″-″　:負の数値の後に−をつけて表示 （書式の右端につける） ″**″　:#で指定した桁数に満たない数値の前を＊で埋める （書式の左端につける） ″¥¥″　:数値の前に¥をつけて表示 （書式の左端につける） ″**¥″　:数値の前に¥をつけて表示し、#で指定した桁数に満たないときは数値の前を＊で埋める （書式の左端につける） ″ ″　:小数点から左に3桁毎に「，」を入れる（整数部の#の並びに置く） ″^^^^″　:数値を指数形式で表示 （書式の右端につける）	文字列や数値を指定した書式でテキスト画面に表示する 〈書式〉は左の書式制御文字で指定する 数値の桁数が〈書式〉で指定した桁数より大きいときは、数値の右側に″%″が表示される 数値の小数部の桁数が指定より大きいときは、四捨五入して表示される	10 A$=″ABCDE″ 20 PRINT USING″!″;A$ 30 PRINT USING″&　&″;A$ 40 PRINT USING″XXX@YYY″;A$ PRINT USING ″###.##″;123.456,789 PRINT USING ″+###″;100,−200 PRINT USING ″###−″;100,−200 PRINT USING ″**###″;10,−20 PRINT USING ″¥¥###″;10,−20 PRINT USING ″**¥###″;10,−20 PRINT USING ″###,.##″;1234.56 PRINT USING ″#.##^^^^″;123.45
PSET [STEP](x,y) [,〈カラーコード〉] [,〈論理演算子〉] (g)	指定された位置に点を描く	PSET (100, 100), 8
PUT KANJI [STEP](x,y) ,〈漢字コード〉[,〈カラーコード〉] [,〈論理演算子〉] [,〈モード〉] (g)	グラフィック画面に漢字を表示する	PUT KANJI (50, 100) , &H3441, 3
PUT SPRITE〈スプライト面番号〉 [STEP] [,(x,y)][,〈カラーコード〉] [,〈スプライトパターン番号〉] (g)	指定されたスプライト面にスプライトパターンを表示する 同じ面に先に表示されていたパターンは消える	PUT SPRITE 0 , (100, 100), 10
SCREEN [〈画面モード〉] [,〈スプライトサイズ〉] [,〈キー音〉] [,〈カセットボーレート〉] [,〈プリンタの種類〉] [,〈表示モード〉]	〈画面モード〉 0〜12 〈スプライトサイズ〉 0:8×8ドット 1:8×8を2倍に拡大 2:16×16ドット 3:16×16を2倍に拡大 〈キー音〉0:なし 1:あり 〈カセットボーレート〉 0:1200 1:2400 〈プリンタの種類〉 0:MSX標準プリンタ 1:非標準プリンタ 〈表示モード〉0〜3	SCREEN 5, 1

＊ＳＥＴ ＰＡＧＥ命令で、ページ０以外のページを描画ページにしてスプライトを使用するときは、次のルーチンを、プログラムのＳＥＴ ＰＡＧＥ命令のすぐ後に実行させてください。

```
100 FOR N=0 TO 31
110 PUT SPRITE N(0,0),0
120 NEXT N
```

書　式	機　能	文　例
ＳＥＴ ＰＡＧＥ [〈表示ページ〉] [,〈描画ページ〉] ＊ g	スクリーン５〜12の画面モードで、表示するＶＲＡＭのページと描画するＶＲＡＭのページを設定する	ＳＥＴ ＰＡＧＥ ０，１
ＳＥＴ ＳＣＲＯＬＬ [〈x〉] [,〈y〉] [,〈目隠し〉] [,〈次ページ〉]	表示画面を水平及び垂直にスクロールさせる	ＳＥＴ ＳＣＲＯＬＬ １２８，１００，０，０
ＷＩＤＴＨ〈桁数〉 t	テキスト画面で１行に表示できる文字数を設定する	ＷＩＤＴＨ ３２

ファイル入出力

（「互換ＲＡＭディスク」とは、従来のＭＳＸ２、２＋ＢＡＳＩＣと互換性のあるＲＡＭディスクのことです。）

書　式	機　能	文　例
ＢＬＯＡＤ ″〈ファイルスペック〉″ [,｜Ｒ／Ｓ｜][,〈オフセット〉]	機械語プログラムやデータをロードする Ｒがついているときはロードしたプログラムを実行する ＳがついているときはＶＲＡＭにデータをロードする オフセットがついているときは、セーブ時の開始アドレスにオフセット分を加えたアドレスにロードする	ＢＬＯＡＤ ″ＰＲＯＧ２．ＢＩＮ″，Ｒ ＢＬＯＡＤ ″ＢＬＵＥ．ＰＩＣ″，Ｓ
ＢＳＡＶＥ ″〈ファイルスペック〉″ ,〈開始アドレス〉,〈終了アドレス〉 [,｜〈実行開始アドレス〉／Ｓ｜]	〈開始アドレス〉から〈終了アドレス〉までのメモリの内容を、機械語プログラムとしてセーブする 実行開始アドレスを省略すると、〈開始アドレス〉から実行する ＳをつけるとＶＲＡＭの内容をフロッピーディスクにセーブする	ＢＳＡＶＥ ″ＰＲＯＧ３．ＢＩＮ″ ，＆ＨＡ１００，＆ＨＡ２ＦＦ
ＣＬＯＳＥ [[＃]〈ファイル番号〉 [,[＃]〈ファイル番号〉…]]	ファイルを閉じる	ＣＬＯＳＥ ＃１，２
ＦＩＥＬＤ [＃]〈ファイル番号〉 ,〈文字長〉 ＡＳ 〈文字型変数〉 [,〈文字長〉 ＡＳ 〈文字型変数〉…]	ランダムアクセスファイルの１レコードを分割する （合計 256文字またはレコード長以内）	ＦＩＥＬＤ ＃１，１６ ＡＳ Ａ＄ ，４ ＡＳ Ｂ＄，２４ ＡＳ Ｃ＄
ＧＥＴ [＃]〈ファイル番号〉 [,〈レコード番号〉]	ランダムアクセスファイルから１レコード入力する	ＧＥＴ ＃１，２
ＩＮＰＵＴ ＃〈ファイル番号〉 ,〈変数名〉[,〈変数名〉…]	シーケンシャルファイルからデータを順に読み取り、変数に入れる	ＩＮＰＵＴ ＃１，Ａ，Ｂ＄，Ｃ
ＬＩＮＥ ＩＮＰＵＴ ＃〈ファイル番号〉 ,〈文字型変数名〉	１行単位のデータをシーケンシャルファイルから読み取り、文字型変数に入れる	ＬＩＮＥ ＩＮＰＵＴ ＃１，Ａ＄
ＬＯＡＤ ″〈ファイルスペック〉″ [,Ｒ]	ＢＡＳＩＣのプログラムをロードする Ｒをつけるとロード後実行	ＬＯＡＤ ″ＰＲＯＧ４″，Ｒ
ＬＳＥＴ 〈文字型変数名〉＝〈文字列〉	文字列をＦＩＥＬＤ文で指定した変数に左づめで入れる	ＬＳＥＴ Ａ＄＝″タンカ″ ＬＳＥＴ Ｂ＄＝Ｄ＄

書　式	機　能	文　例
MAXFILES=〈ファイル数〉	同時にOPENする最大のファイル数を宣言する	MAXFILES=2
MERGE "〈ファイルスペック〉"	アスキーセーブしたプログラムをロードしてメモリ内のプログラムと混合する	MERGE "PROG5.ASC"
OPEN "〈ファイルスペック〉" [FOR 〈モード〉] AS #〈ファイル番号〉 [LEN=〈レコード長〉]	ファイルを開く 〈モード〉 ・シーケンシャルファイル OUTPUT…出力用 INPUT……入力用 APPEND…追加用 なし……テキスト画面 グラフィック画面 またはプリンタ ・ランダムファイル…なし	OPEN "FILE1.DAT" AS #1 グラフィック画面 OPEN "GRP:" FOR OUTPUT AS #1
PRINT #〈ファイル番号〉 [{,|;}〈式または文字列〉 [{,|;}〈式または文字列〉]…]	シーケンシャルファイルにデータを出力する	PRINT #1, A, B, C
PRINT #〈ファイル番号〉 USING 〈書式〉;〈式または文字列〉 [[{,|;}〈式または文字列〉]…]	指定された〈書式〉でシーケンシャルファイルにデータを出力する	PRINT #1 USING "###.##";A$, B;C$
PUT [#]〈ファイル番号〉 [,〈レコード番号〉]	ランダムアクセスファイルに1レコード出力する	PUT #1, 1
RSET 〈文字型変数名〉=〈文字列〉	文字列をFILED文で指定した変数に右詰めで入れる	RSET A$="すうりょう" RSET B$=D$
SAVE "〈ファイルスペック〉" [,A]	BASICのプログラムをセーブする デバイス名がドライブ名または省略されたときは、通常の形式でディスクにセーブ 末尾にAを入力されたときは、常にアスキーセーブ	SAVE "PROG7.ASC" SAVE "PROG7.ASC", A

〈ファイルスペック〉の形式

"[〈デバイス名〉:] [〈パスリスト〉] [〈ファイル名〉] [.〈拡張子〉]"

・**〈デバイス名〉**:は次のとおり。

CRT:……テキスト画面
GRP:……グラフィック画面
LPT:……プリンタ
MEM:……互換RAMディスク
A:〜G:…・フロッピーディスクのドライブ名
H:…………RAMディスク
省略…………A:を指定したときと同じ

・**〈パスリスト〉**は、デバイスがフロッピーディスクまたはRAMディスクのときのみ使用可能。
・**〈ファイル名〉**は、デバイスがフロッピーディスクまたはRAMディスクのときは8文字以内。それ以外は省略する。
・**〈拡張子〉**はデバイスがフロッピーディスクまたはRAMディスクのときに3文字以内でつける。それ以外は省略する。

（「互換ＲＡＭディスク」とは、従来のＭＳＸ２、ＭＳＸ２＋ＢＡＳＩＣと互換性のあるＲＡＭディスクのことです。　）

ファイル管理

書　式	機　能	文　例
ＣＡＬＬ　ＦＯＲＭＡＴ	ディスクをフォーマットする	ＣＡＬＬ　ＦＯＲＭＡＴ
ＣＡＬＬ　ＣＨＤＩＲ （″[〈ドライブ名〉][〈パスリスト〉] [〈ディレクトリ名〉]″）	カレントディレクトリを変更または表示する	ＣＡＬＬ　ＣＨＤＩＲ（″ＷＯＲＤＳ″） ＣＡＬＬ　ＣＨＤＩＲ（″Ａ：″）
ＣＡＬＬ　ＣＨＤＲＶ （″[〈ドライブ名〉]″）	カレントドライブを変更する	ＣＡＬＬ　ＣＨＤＲＶ（″Ｈ：″）
ＣＡＬＬ　ＭＥＭＩＮＩ [(〈互換ＲＡＭディスクの上限〉)]	メモリ内に互換ＲＡＭディスク領域を設定し、初期化する	ＣＡＬＬ　ＭＥＭＩＮＩ　（３０００）
ＣＡＬＬ　ＭＦＩＬＥＳ	互換ＲＡＭディスク内のファイルスペックを表示する	ＣＡＬＬ　ＭＦＩＬＥＳ
ＣＡＬＬ　ＭＫＤＩＲ （″[〈ドライブ名〉][〈パスリスト〉] [〈ディレクトリ〉]″）	新しいサブディレクトリを作成する	ＣＡＬＬ　ＭＫＤＩＲ（″ＷＯＲＤＳ″） ＣＡＬＬ　ＭＫＤＩＲ （″ＷＯＲＤＳ￥ＴＥＸＴ″）
ＣＡＬＬ　ＭＫＩＬＬ （″〈ファイルスペック〉″）	互換ＲＡＭディスク内のファイルを消去する	ＣＡＬＬ　ＭＫＩＬＬ （″ＰＲＯＧ８．ＢＡＳ″）
ＣＡＬＬ　ＭＮＡＭＥ （″〈旧ファイルスペック〉″ ＡＳ　″〈新ファイルスペック〉″）	互換ＲＡＭディスク内のファイルスペックを変更する	ＣＡＬＬ　ＭＮＡＭＥ　（″ＰＲＯＧ９″ ＡＳ　″ＰＲＯＧ９．ＢＡＳ″）
ＣＡＬＬ　ＲＡＭＤＩＳＫ [([〈容量〉][,〈変数〉])]	メモリ内にＲＡＭディスクの領域を設定し、設定できた容量を変数の値にする	ＣＡＬＬ　ＲＡＭＤＩＳＫ（６４） ＣＡＬＬ　ＲＡＭＤＩＳＫ （１０００，Ａ）：？Ａ
ＣＡＬＬ　ＲＭＤＩＲ （″[〈ドライブ名〉][〈パスリスト〉] [〈ディレクトリ名〉]″）	ひとつまたは複数の空のサブディレクトリを削除する	ＣＡＬＬ　ＲＭＤＩＲ（″ＮＥＸＴ″） ＣＡＬＬ　ＲＭＤＩＲ（″ＷＯＲＤ＊″）
ＣＯＰＹ　″〈ファイルスペック〉″ ＴＯ　″〈ファイルスペック〉″	ディスク内のファイルを同じディスクまたは他のディスクにコピーする	ＣＯＰＹ″Ａ：ＷＯＲＤＳ￥ＭＩＫＥ″ ＴＯ″Ｂ：ＬＴＲＳ￥ＴＡＭＡ″
ＦＩＬＥＳ　[″〈ファイルスペック〉″]	ディスク内のファイルスペックを表示する	ＦＩＬＥＳ ＦＩＬＥＳ　″＊．ＢＡＳ″
ＫＩＬＬ　″〈ファイルスペック〉″	ディスク内のファイルを消去する	ＫＩＬＬ　″ＰＲＯＧ１０．ＢＡＳ″
ＬＦＩＬＥＳ　[″〈ファイルスペック〉″]	ディスク内のファイルスペックをプリンタで印刷する	ＬＦＩＬＥＳ
ＮＡＭＥ　″〈旧ファイルスペック〉″ ＡＳ　″〈新ファイルスペック〉″	ディスク内のファイルスペックを変更する	ＮＡＭＥ　″ＰＲＯＧ１．ＢＡＳ″ ＡＳ″ＧＡＭＥ１．ＢＡＳ″

割り込み

書　式	機　能	文　例
ＩＮＴＥＲＶＡＬ　{ＯＮ / ＯＦＦ / ＳＴＯＰ}	タイマー割り込みを制御 ＯＮ　：許可する ＯＦＦ　：禁止する ＳＴＯＰ：保留する	ＩＮＴＥＲＶＡＬ　ＯＮ ＩＮＴＥＲＶＡＬ　ＯＦＦ ＩＮＴＥＲＶＡＬ　ＳＴＯＰ
ＫＥＹ(〈キー番号〉)　{ＯＮ / ＯＦＦ / ＳＴＯＰ}	ファンクションキー割り込みを制御 ＯＮ　：許可する ＯＦＦ　：禁止する ＳＴＯＰ：保留する	ＫＥＹ（１）　ＯＮ ＫＥＹ（２）　ＯＦＦ ＫＥＹ（３）　ＳＴＯＰ

書　式	機　能	文　例
ON INTERVAL=〈時間〉 GOSUB 〈行番号〉	タイマー割り込みの条件を設定し、割り込み処理ルーチンの開始行を定義する	ON INTERVAL=60 GOSUB 100
ON KEY GOSUB [〈行番号〉] [,〈行番号〉…]	ファンクションキー割り込み処理ルーチンの開始行を定義	ON KEY GOSUB 100, 200, 300
ON SPRITE GOSUB 〈行番号〉	スプライトが重なったときに実行する割り込み処理ルーチンの開始行を定義する	ON SPRITE GOSUB 100
ON STOP GOSUB 〈行番号〉	CTRL + STOP 割り込み処理ルーチンの開始行を定義	ON STOP GOSUB 100
ON STRIG GOSUB [〈行番号〉] [,〈行番号〉…]	SPACE やトリガーボタンによる割り込みの処理ルーチンの開始行を定義する	ON STRIG GOSUB 100, 200
SPRITE {ON / OFF / STOP}	スプライトが重なったときの割り込みを制御 ON　：許可する OFF　：禁止する STOP：保留する	SPRITE ON SPRITE OFF SPRITE STOP
STOP {ON / OFF / STOP}	CTRL + STOP 割り込みの制御 ON　：許可する OFF　：禁止する STOP：保留する	STOP ON STOP OFF STOP STOP
STRIG (〈ジョイスティックの番号〉) {ON / OFF / STOP}	トリガーボタンによる割り込みを制御 ON　：許可する OFF　：禁止する STOP：保留する	STRIG ON STRIG OFF STRIG STOP

I／Oポート、メモリ関係

書　式	機　能	文　例
OUT 〈ポート番号〉,〈式〉	I／Oポートに1バイトのデータを出力する	OUT &H22, &H80
POKE 〈アドレス〉,〈データ〉	メモリ内の指定されたアドレスにデータを書き込む	POKE &HF000, &HFF
VPOKE 〈アドレス〉,〈データ〉	VRAM内の指定されたアドレスにデータを書き込む	VPOKE &H8000, &HFF
WAIT 〈ポート番号〉,〈式1〉 [,〈式2〉]	I／Oポートからある値のデータが入力するまでプログラムの実行を中断する	WAIT 1, &H22, &H22

接続機器

書　式	機　能	文　例
LPRINT [〈式〉]	文字列または数値をプリンタで印刷する	LPRINT "123";A, B$
LPRINT USING 〈書式〉 ;〈式〉 [{, / ;} 〈式〉]…	文字列または数値を指定された書式でプリンタで印刷する（〈書式〉はPRINT USINGを参照）	LPRINT USING "###" ;A, B
LLIST [〈開始行番号〉] [−] [〈終了行番号〉]	プログラムのリストをプリンタで印刷する	LLIST 100−200

資料編

音に関する命令

書式	機能	文例
BEEP	ビープ音を鳴らす	BEEP
CALL BGM (n)	FM音源の演奏のバックグラウンド処理を指定する nは、0：指定、1：取消	CALL BGM (0) CALL BGM (1)
CALL MUSIC [(〈モード〉[,0 [,〈n1〉 [,〈n2〉 [,… … [,〈n9〉]]]]]]]]]])]	FM音源用拡張BASICをスタートさせ、各種の初期設定をする 〈モード〉0：リズム音なし 1：リズム音あり 〈n1〉～〈n9〉： PLAY#文の各文字列に対するチャンネル数	CALL MUSIC CALL MUSIC (1, 0, 2, 2, 2)
CALL PCMPLAY (@〈開始番地〉,〈終了番地〉 ,〈細かさ〉 [,s]) CALL PCMPLAY (@〈配列変数名〉,[〈長さ〉] ,〈細かさ〉)	PCM音源で録音された音声を再生する 〈細かさ〉は0～3の数値 「, s」ありの場合VRAMに記録したデータを再生し、なしの場合メインメモリに記録したデータを再生する 配列変数からデータを再生するとき、〈長さ〉を省略すると配列全体の内容を再生する	CALL PCMPLAY (@&HB000, &HDFFF, 3) CALL PCMPLAY (@A, , 3)
CALL PCMREC (@〈開始番地〉, 〈終了番地〉, 〈細かさ〉 [, [〈音量〉] , 〈圧縮指定〉]) CALL PCMREC (@〈開始番地〉, 〈終了番地〉, 〈細かさ〉, [〈音量〉] , [〈圧縮指定〉] , s) CALL PCMREC (@〈配列変数名〉, [〈長さ〉] , 〈細かさ〉 [,[〈音量〉],〈圧縮指定〉])	PCM音源で音声を録音する 〈細かさ〉は0～3、 〈音量〉は0～127の数値 〈圧縮指定〉は、 0または省略：圧縮しない 1のとき　：圧縮する 「, s」なしの場合、録音した音声データをメインメモリに記録し、「, s」ありの場合、VRAMに記録する 録音した音声データを配列変数に記録する場合、〈長さ〉を省略すると配列全体に記録する	CALL PCMREC (@&HD000, &HE500, 3) CALL PCMREC (@&H1000, &H2000 , 2, 8, 1, s) CALL PCMREC (@A, , 1)
CALL PITCH (〈周波数〉)	FM音源の全楽音の音の高さの基準を設定・変更する	CALL PITCH (450)
CALL PLAY (〈チャンネル番号〉, 〈変数〉)	FM音源の指定されたチャンネルが演奏中なら変数の値を－1に、停止中なら0にする	CALL PLAY (1, A)
CALL STOPM	FM音源のバックグラウンド演奏を中止する	CALL STOPM
CALL TEMPER (〈音律番号〉)	FM音源の楽音の音律を指定する (0～21)	CALL TEMPER (0)
CALL TRANSPOSE (〈セント〉)	FM音源が演奏する音を半音の1/100単位で移調する	CALL TRANSPOSE (100)

書　式	機　能	文　例
CALL VOICE ([@n1] , [@n2] ,… [@n9])	FM音源の、楽音を演奏するチャンネルに音色を指定する（n1～n9は音色番号）	CALL VOICE (@0, @2, @3)
CALL VOICE COPY (@n1, @n2)	FM音源の音色データを音色番号63（無音）または配列変数にコピーする n1は0～63（音色番号） n2は63または配列変数名	CALL VOICE COPY (@0, @63) CALL VOICE COPY (@2, A)
PLAY ″〈文字列1〉″ [,″〈文字列2〉″] [,″〈文字列3〉″]	PSGで音楽を演奏する （〈文字列〉はPSG用MML）	PLAY ″O4CDEFGABO5C″ PLAY A$, B$+C$
PLAY [#〈モード〉,] [0] , [″〈FM音源用文字列1〉″ [, ″〈FM音源用文字列2〉″ …. [, ″〈FM音源用文字列n〉″ [, ″〈リズム音用文字列〉″ [, ″〈PSG用文字列1〉″ …. [, ″〈PSG用文字列3〉″]]]]]]]]]]]]]]	FM音源とPSGで演奏する 〈モード〉 0または省略のとき： PSGのみ演奏可能 〈FM音源用文字列〉省略 〈リズム音用文字列〉省略 2または3のとき　： FM音源とPSG演奏可能 〈FM音源用文字列〉の数はMUSIC文の指定以内 〈リズム音用文字列〉はMUSIC文のモード指定が「リズム音なし」の時省略	PLAY #2, 0 , ″CDE″, ″EFG″, ″GAB″ , ″B4S4H4!″, ″O3DEF″
SET BEEP〈音色の種類〉[,〈音量〉]	ビープ音の種類と音量を設定	SET BEEP 1, 3
SOUND 〈レジスタ番号〉,〈データ〉	PSG（音源IC）のレジスタに値を直接書き込む	SOUND 7, 254:SOUND 8, 8:SOUND 0, 255

●**楽音用ミュージック・マクロ・ランゲージ(MML)**（＊1はPSG専用。＊2はMUSIC文実行後のみ有効。）

MML	機能と設定範囲	初期値
A[n]～G[n]	音符(ド～シがC,D,E,F,G,A,Bに対応)nは音符の長さ	――――
Ln	省略時の音の長さを設定　1≦n≦64	n=4
Mn	音色のエンベロープ周期の設定　1≦n≦65535　＊1	――――
Nn	音の高さを細かく設定　0≦n≦96	――――
On	オクターブを設定　1≦n≦8	n=4
Qn	音の長さの割合　1≦n≦8　＊2	n=8
R[n]	休符　1≦n≦64	n=4
Sn	音色のエンベロープ形状の設定　0≦n≦15　＊1	――――
Tn	テンポの設定　32≦n≦255	n=120
Vn	音量の設定　0≦n≦15	n=8
XA$;	文字型変数に入っているMMLを実行	――――
=x;	数値型変数の値を直前のMMLのnにする	――――
.	直前の音符または休符の長さを1.5倍に（付点音符）	――――
#, +	直前の音符を半音上げる（シャープ）	――――
−	直前の音符を半音下げる（フラット）	――――
〉	オクターブを1つ上げる　＊2	――――
〈	オクターブを1つ下げる　＊2	――――
&	タイ　前後の音をつなぐ　＊2	――――
{ } n	連符　n分音符を { } 中の音程の個数で等分にした音を発生する　1≦n≦64　＊2	Lnで設定された値
@n	n番の音色に切り替える　0≦n≦63　＊2	n=0
@Vn	音量を細かく設定する　0≦n≦127　＊2	――――
@Wn	nで指定された長さだけ状態を維持　1≦n≦64　＊2	Lnで設定された値

●**リズム音用ミュージック・マクロ・ランゲージ(MML)**

MML	機能と設定範囲	設定範囲	初期値
B	バス・ドラム音を発声	―	―
S	スネア・ドラム音を発声	―	―
M	タムタム音を発声	―	―
C	シンバル音を発声	―	―
H	ハイハット音を発声	―	―
n(数値)	直前までの楽音を発声し、n分音符分待つ	1 ≦ n ≦ 64	―

(FM音源専用; 3音まで同時発声可能)

MML	機能と設定範囲	設定範囲	初期値
Vn	音量を設定する	0 ≦ n ≦ 15	8
!	直前の楽音をアクセントボリュームにする	―	―
@An	アクセントボリュームの音量を設定する	0 ≦ n ≦ 15	―

「Tn」、「@Vn」、「Rn」、「XA$;」、「=x;」、「.」(ピリオド)は楽音用と同じ。

●**音色データ**(FM音源専用:VOICE文およびMMLの@nで使える音色データの一覧表)

音色番号	音色名	音色番号	音色名	音色番号	音色名
0*	ピアノ1	22	チャイム	44	チャーチオルガンR
1	ピアノ2	23*	シンセ・ベース	45	シンセ・バイオリン
2*	バイオリン	24*	シンセサイザー	46	シンセ・オルガン
3*	フルート1	25	シンセ・ドラム	47	シンセ・ブラス
4*	クラリネット	26	シンセ・リズム	48*	ホルン
5*	オーボエ	27	ハーモ・ドラム	49	三味線
6*	トランペット	28	カウベル	50	マジカル
7	パイプオルガン1	29	ハイハット	51	フワワ
8	シロフォン	30	スネア・ドラム	52	ワンダーフラット
9*	オルガン	31	バス・ドラム	53	ハードロック
10*	ギター	32	ピアノ3	54	マシーン
11	サンツール1	33*	ウッドベース	55	マシーンV
12*	エレキベース	34	サンツール2	56	コミック
13	クラビコード1	35	ブラス	57	SE−コミック
14*	ハープシコード1	36	フルート2	58	SE−レーザー
15	ハープシコード2	37	クラビコード2	59	SE−ノイズ
16*	ビブラフォン	38	クラビコード3	60	SE−星1
17	琴1	39	琴2	61	SE−星2
18	太鼓	40	パイプオルガン2	62	エンジン2
19	エンジン1	41	PohdsPLA	63	無音
20	UFO	42	PohdsPRA		
21	シンセサイザベル	43	チャーチオルガンL		

*印はFM音源が内蔵する音色。他は内蔵プログラムにより合成される音色で、他の音色と同時に発声できない。
音色によっては、実際の楽器の音色と異なるものもある。

●**音律一覧表**(FM音源専用:TEMPER文で使える音律の一覧表)

番号	音律	番号	音律
0	ピタゴラス	11	純正律 cisメジャー (b マイナー)
1	ミーントーン	12	純正律 dメジャー (h マイナー)
2	ヴェルクマイスター	13	純正律 esメジャー (c マイナー)
3	ヴェルクマイスター(修正)*1	14	純正律 eメジャー (cis マイナー)
4	ヴェルクマイスター(別)	15	純正律 fメジャー (d マイナー)
5	キルンベルガー	16	純正律 fisメジャー (es マイナー)
6	キルンベルガー8修正)*1	17	純正律 gメジャー (e マイナー)
7	ヴァロッティ・ヤング	18	純正律 gisメジャー (f マイナー)
8	ラモー	19	純正律 aメジャー (fis マイナー)
9	完全平均律(初期値)	20	純正律 bメジャー (g マイナー)
10	純正律 cメジャー (a マイナー)	21	純正律 hメジャー (gis マイナー)

*1:修正は、平島達司氏によるもの。

エラー処理

書式	機能	文例
ERROR 〈エラー番号〉	指定したエラー番号のエラーを発生する	ERROR 3
ON ERROR GOTO 〈行番号〉	エラーが発生したら、指定した行番号にジャンプする	ON ERROR GOTO 1000
RESUME [0 / NEXT / 〈行番号〉]	エラー処理ルーチンからメインプログラムに戻る 0および省略はエラーが発生した文に NEXTはその次の文に	RESUME 10

その他の命令

書式	機能	文例
CALL 〈拡張命令〉	拡張命令を呼び出す	CALL MUSIC _MUSIC
CALL SYSTEM [(″DOSコマンド″)]	MSX−DOS2に制御を移す	CALL SYSTEM _SYSTEM
GET DATE 〈文字型変数〉 GET TIME 〈文字型変数〉	日付、時刻を変数に入れる	GET DATE D$ GET TIME T$
SET ADJUST (x,y)	画面の表示位置を調整する (x,yはそれぞれ−7〜8)	SET ADJUST (−3, 5)
SET PASSWORD ″〈パスワード〉″	MSXのタイトル画面にパスワードを設定する 次回から、パスワードを入力するまで待ち続ける パスワードを忘れたときは次のようにする 1.内蔵ソフト切り替えスイッチがあるパソコンでは、あらかじめ「切」にする 2. GRAPH と STOP を押しながらリセットボタンを押し、BASICの画面が表示されるまで押し続ける 3. STOP を押してキー入力を可能にする 4.下の文を入力する SET TITLE ″″	SET PASSWORD ″AAA″
SET PROMPT ″〈プロンプト〉″	ダイレクトモードの「Ok」を他の文字に変更する	SET PROMPT ″Ready!″
SET TITLE ″〈タイトル〉″ [,〈背景色〉]	MSXのタイトル画面の背景色(1〜4)とMSXマークの下に表示される〈タイトル〉を指定する	SET TITLE ″Hello″, 2
SET SCREEN	実行時のSCREEN命令の設定をSRAMに記録する 次回から同じ設定でスタート	SET SCREEN
SET DATE ″〈日付〉″ SET TIME ″〈時刻〉″	日付や時刻を設定する	SET DATE ″89/01/15″ SET TIME ″12:34:56″
SWAP 〈変数〉,〈変数〉	2つの変数の値を交換する	SWAP A, B

関　数

以下、nは数値、数値型変数または計算式をあらわします。

数値型関数

書　式	機　能	文　例
ABS (n)	絶対値を返す	A=ABS (B)
ATN (n)	アークタンジェントを返す (nはπを単位とする角度)	A=ATN (B/180*3.1415)
CALL KINSTR (〈数値型変数〉 [,n],〈文字列1〉,〈文字列2〉) **注：漢字モードでのみ使用可能**	〈文字列1〉のn文字目から後ろの〈文字列2〉の位置を、見つからなければ0を変数に入れる nの省略時は1文字目から	CALL KINSTR (A, 10, B$, "愛")
CALL KLEN (〈数値型変数〉,〈文字列〉[,〈機能〉]) **注：漢字モードでのみ使用可能**	〈機能〉が0（または省略）のときは、〈文字列〉の全体の長さを、1のときは半角文字の数を、2のときは全角文字の数を変数に入れる	CALL KLEN (A, B$, 2)
CALL KTYPE (〈数値型変数〉,〈文字列〉, n) **注：漢字モードでのみ使用可能**	〈文字列〉中のn番目の文字のタイプ（半角なら0，全角なら1）を変数に入れる	CALL KTYPE (A, B$, 1)
CDBL (n)	倍精度実数に変換して返す	A=CDBL (B%)
CINT (n)	整数に変換して返す	A%=CINT (B)
COS (n)	コサインを返す (nはラジアン単位の角度)	A=COS (B/180*3.1415)
CSNG (n)	単精度実数に変換して返す	A!=CSNG (B)
ERL	発生したエラーのエラー番号を返す	A=ERL IF A=68 THEN 100
ERR	エラーが発生した行の行番号を返す	A=ERL IF A=40 THEN 200
EXP (n)	e^nを返す	A=EXP (B)
FIX (n)	nの整数部分を返す	A=FIX (B)
INT (n)	n未満の最大の整数を返す	A=INT (B)
LOG (n)	自然対数を返す	A=LOG (B)
RND [(n)]	0以上1未満の乱数を返す nが0（または省略）のとき直前に発生した乱数の値、nが正のときはRUN文を実行するたびに同じ系列の乱数、nが負のときはnの値によって決まった値を発生。負のときは同時に乱数系列を初期化	A=RND (−B)
SGN (n)	nが正なら1、負なら−1、0なら0を返す	A=SGN (B)
SIN (n)	サインを返す (nはラジアン単位の角度)	A=SIN (B/180*3.1415)
SQR (n)	平方根を返す	A=SQR (B)
TAN (n)	タンジェントを返す (nはラジアン単位の角度)	A=TAN (B)

文字型関数

書　式	機　能	文　例
CALL AKCNV (〈文字型変数〉,〈文字列〉) **注：**漢字モードでのみ使用可能	〈文字列〉のすべての文字を全角文字に変換し、〈文字型変数〉に入れる	CALL　AKCNV（A$, B$）
CALL JIS（〈文字型変数〉,〈文字列〉） **注：**漢字モードでのみ使用可能	〈文字列〉の最初の2バイトをJISコードに変換して〈文字型変数〉に入れる	CALL　JIS（A$, B$）
CALL SJIS (〈文字型変数〉,〈文字列〉) **注：**漢字モードでのみ使用可能	〈文字列〉の最初の2バイトをシフトJISコードに変換して〈文字型変数〉に入れる	CALL　SJIS（A$, B$）
CALL　KACNV (〈文字型変数〉,〈文字列〉) **注：**漢字モードでのみ使用可能	〈文字列〉の半角文字に変換できる文字をすべて半角文字に変換し、〈文字型変数〉に入れる	CALL　KACNV（A$, B$）
CALL　KEXT (〈文字型変数〉,〈文字列〉,〈機能〉) **注：**漢字モードでのみ使用可能	〈機能〉が0なら〈文字列〉の半角文字を、1なら全角文字を〈文字型変数〉に入れる	CALL　KEXT（A$, B$, 1）
CALL　KMID (〈文字型変数〉〈文字列〉, n1 [, n2]) **注：**漢字モードでのみ使用可能	〈文字列〉の左からn1番目の文字からn2個の文字を〈文字型変数〉に入れる （n2省略時は終わりまで）	CALL　KMID（A$, B$, 10, 3）
CALL　KNJ (〈文字型変数〉,"〈漢字コード〉") **注：**漢字モードでのみ使用可能	「"」で囲んだ4桁の数字で指定される漢字コードに相当する全角文字を〈文字型変数〉に入れる	CALL　KNJ（A$, "3441"） CALL　KNJ（A$, B$）
LEFT$（〈文字列〉, n）	〈文字列〉の左からn個の文字を返す	A$=LEFT$（B$, 3）
MID$（〈文字列〉, n1 [, n2]）	〈文字列〉の左からn1番目の文字からn2個の文字を返す （n2省略時は終わりまで）	A$=MID$（B$, 5, 1）
RIGHT$（〈文字列〉, n）	〈文字列〉の右からn個の文字を返す	A$=MID$（B$, 3）
SPACE$（n）	n個の空白を文字列として返す	A$=SPACE$（12）
SPC（n）	n個の空白を出力する	PRINT　SPC（12）
STRING$ （n, |〈文字列〉 　　|〈キャラクタコード〉|）	〈文字列〉の最初の文字または〈キャラクタコード〉（0～255）であらわされる文字をn個分出力する	STRING$（10, "+"）
TAB（n）	指定された桁まで空白を出力	TAB（10）

数値型⇆文字型変換に関する関数

書　　式	機　能	文　例
ASC（〈文字列〉）	〈文字列〉の先頭の文字のキャラクタコードを返す	A＝ASC（B$）
BIN$（n）	nの2進表記を文字列として返す	A$＝BIN$（15）
CALL KINSTR（〈数値型変数〉 [,n],〈文字列1〉,〈文字列2〉） 注：漢字モードでのみ使用可能	〈文字列1〉のn文字目から後ろの〈文字列2〉の位置を、見つからなければ0を変数に入れる nの省略時は1文字目から	CALL KINSTR （A, 10, B$, "愛"）
CALL KLEN （〈数値型変数〉,〈文字列〉[,〈機能〉]） 注：漢字モードでのみ使用可能	〈機能〉が0（または省略）のときは、〈文字列〉の全体の長さを、1のときは半角文字の数を、2のときは全角文字の数を変数に入れる	CALL KLEN（A, B$, 2）
CALL KTYPE （〈数値型変数〉,〈文字列〉, n） 注：漢字モードでのみ使用可能	〈文字列〉中のn番目の文字のタイプ（半角なら0，全角なら1）を変数に入れる	CALL KTYPE（A, B$, 1）
CHR$（n）	nをキャラクタコードとしてもつ文字を返す	A$＝CHR$（B）
CVI CVS CVD	ランダムアクセスファイルから入力した文字型データを数値に変換して返す	A＝CVI（A$） B＝CVS（B$） C＝CVD（C$）
MKI$ MKS$ MKD$	ランダムアクセスファイルに保存するため、数値を文字型データに変換する	A$＝MKI$（A） B$＝MKS$（B） C$＝MKD$（C）
HEX$（n）	nの16進表記を文字列として返す	A$＝HEX$（15）
INSTR（[n,]〈文字列1〉,〈文字列2〉）	〈文字列1〉のn個目の文字から後ろで〈文字列2〉を探し、その位置を返す	INSTR（2, A$, B$）
LEN（〈文字列〉）	〈文字列〉の文字数を返す	A＝LEN（B$）
OCT$（n）	nの8進表記を文字列として返す	A$＝OCT$（15）
STR$（n）	数値をあらわす文字列を返す	A$＝STR$（20）
VAL（〈文字列〉）	文字列があらわす数値を返す	A＝VAL（"15"）

データの入力に関する関数

●スクリーンからの入力

書　　式	機　能	文　例
CSRLIN　（システム変数）	カーソルのタテ座標を返す	A＝CSRLIN
POS（ダミーの数値）　（システム変数）	カーソルのヨコ座標を返す	A＝POS（0）
POINT（x, y）	（x, y）の位置の点のカラーコードを返す	A＝POINT（50, 100）

●データファイルからの入力

書　式	機　能	文　例
ＥＯＦ（〈ファイル番号〉）	ファイルの最後のデータがすでに入力されていたら－１それ以外は０を返す	Ａ＝ＥＯＦ（１） ＩＦ　Ａ＝－１　ＴＨＥＮ…
ＩＮＰＵＴ＄（ｎ［,［＃］〈ファイル番号〉］）	ファイルからｎ文字入力して返す	Ａ＄＝ＩＮＰＵＴ＄（６，＃１）
ＬＯＦ（〈ファイル番号〉）	ファイルの長さをバイト単位で返す	Ａ＝ＬＯＦ（１）
ＬＯＣ（〈ファイル番号〉）	ランダムアクセスファイルのときは、現在のレコード番号を返す シーケンシャルファイルのときは、ＯＰＥＮしてから入出力したレコード数（ＩＮＰＵＴモードではレコード数＋１）を返す。	Ａ＝ＬＯＣ（１）

●プリンタからの入力

書　式	機　能	文　例
ＬＰＯＳ（〈ダミーの数値〉）	プリンタバッファ内のプリンタヘッドの位置を返す	Ａ＝ＬＰＯＳ（０）

●メモリからの入力

書　式	機　能	文　例
ＦＲＥ（｛〈ダミーの数値〉 〈ダミーの文字列〉｝）	引数が〈ダミーの数値〉のときは、メモリの未使用領域の大きさを返す 〈ダミーの文字列〉のときは、メモリの文字列領域の未使用部分の大きさを返す	Ａ＝ＦＲＥ（０） Ａ＝ＦＲＥ（“”）
ＰＥＥＫ（〈アドレス〉）	メモリの指定されたアドレスの内容を返す	Ａ＝ＰＥＥＫ（＆ＨＡ０００）
ＶＡＲＰＴＲ（〈変数名〉）	指定された変数の値が格納されている先頭アドレスを返す	Ａ＝ＶＡＲＰＴＲ（Ｂ）
ＶＡＲＰＴＲ（＃〈ファイル番号〉）	指定されたファイルのファイルコントロールブロックの先頭アドレスを返す	Ａ＝ＶＡＲＰＴＲ（＃１）
ＶＰＥＥＫ（〈アドレス〉）	ＶＲＡＭの指定されたアドレスの内容を返す	Ａ％＝ＶＰＥＥＫ（＆Ｈ１０００）

●キーボードからの入力

書　式	機　能	文　例
ＩＮＫＥＹ＄	現在、押されているキーの文字を返す（何も押されていなければ“”を返す）	Ａ＄＝ＩＮＫＥＹ＄
ＩＮＰＵＴ＄（ｎ）	キーボードからｎ文字入力する	Ａ＄＝ＩＮＰＵＴ＄（３）

●ディスクからの入力

書式	機能	文例
DSKF（〈ドライブ番号〉）	番号で指定されたドライブ内のディスクの残り容量をクラスタ単位で返す 0と1…ドライブA： 2………ドライブB： ・ ・ 8………ドライブH：	PRINT　DSKF（1）

●I／Oポートからの入力

書式	機能	文例
INP（〈ポート番号〉）	I／Oポートから入力しているデータを返す	A＝INP（15）

●機械語サブルーチンからの入力

書式	機能	文例
USR［n］（〈引数〉）	指定された番号（0～9）のユーザーサブルーチンを実行し、結果を返す	A＝USR　1（2）

●タッチパッド（マウス）、ジョイスティック（ジョイパッド）からの入力

書式	機能	文例
PAD（n）	タッチパッドなどの状態を返す（マウスの場合のみ説明） n＝12または16のとき 　マウスに対する入力要求 　（必ず－1を返す） n＝13または17のとき 　マウスのヨコ座標 n＝14または18のとき 　マウスのタテ座標 （トラックボールも同じ）	A＝PAD（12）
STICK（n）	nで指定されたジョイスティックの方向を数値で返す （nが0ならカーソルキー） 0：中央 1：上　　5：下 2：右上　6：左下 3：右　　7：左 4：右下　8：左上	A＝STICK（0）
STRIG（n）	nで指定されたジョイスティックなどのトリガーボタンが押されていれば1を、押されていなければ0を返す （nが0なら SPACE キー）	A＝STRIG（0）

定数と変数

定数	文字型	0～255文字の文字列。「"」（ダブルクォート）で囲む。
	整数型	－32768～＋32767
	固定小数点型	小数点を含む14桁までの数値（実数）。14桁を超える数値の表示は、浮動小数点型になる。
	浮動小数点型	Ｅを使った指数で表示。仮数部の有効数字14桁（倍精度）または６桁（単精度）。 指数部の範囲　－64～＋62
	16進形式	＆Ｈを数字の前につける。
	８進形式	＆Ｏを数字の前につける。
	２進形式	＆Ｂを数字の前につける。

変数	変数名	先頭の２文字のみ有効。（ただし、型宣言文字や配列変数の添え字は変数名に含めない）
	型宣言文字	変数名の後ろにつける。 ％：整数型　　＃：倍精度　　！：単精度　　＄：文字型
	配列変数	変数名の後ろに（　）で囲んで添え字をつける。添え字が10を超える場合は事前に宣言が必要。 添え字は０から始まる整数。各次元の添え字のあいだは「，」で区切る。

特別な変数(システム変数)

書　　式	機　能	文　　例
ＢＡＳＥ（ｎ）	画面出力に関するテーブルのアドレスを返す。ｎは０～44	Ａ＝ＢＡＳＥ（１０）
ＳＰＲＩＴＥ＄＝<文字列>［＋<文字列>…］	スプライトパターンを定義する ８×８ドットのとき‥８文字 16×16ドットのとき‥32文字	SPRITE$=CHR$(24)+CHR$(60)+CHR$(90) +CHR$(126)+CHR$(255)+CHR$(36)+CHR$ (66)+CHR$(66) SPRITE$=""
ＴＩＭＥ	内蔵タイマーの値を示す 数値を入れて書き換え可能	ＰＲＩＮＴ　ＴＩＭＥ ＴＩＭＥ＝０
ＶＤＰ（<レジスタ番号>）	ＶＤＰのレジスタを直接参照更新する	Ａ＝ＶＤＰ（８）

演算記号

算術演算記号	^	累乗（べき）	ＭＯＤ	整数剰余
	－	正負符号を反転	＋，－	加算、減算
	＊，／	乗算、除算	（上から順に優先。ただし式中に「かっこ」があるときは、内側のかっこの中から先に演算する）	
	￥	整数除算		
関係演算記号	＞，＜，＝	大小比較。関係が真のときは演算結果の値が－１、偽のときは値が０になる		
論理演算記号	ＮＯＴ	否定	ＸＯＲ	排他的論理和
	ＡＮＤ	論理積	ＥＱＶ	排他的論理和の否定
	ＯＲ	論理和	ＩＭＰ	含意

カラーコード(初期設定時)

コード	色	コード	色	コード	色	コード	色
0	透　明	4	濃い青	8	赤	12	濃い緑
1	黒	5	青	9	淡い赤	13	紫
2	緑	6	濃い赤	10	黄	14	灰
3	淡い緑	7	水　色	11	淡い黄	15	白

BASIC予約語表

ABS
AND
ASC
ATN
ATTR$
AUTO
BASE
BEEP
BIN$
BLOAD
BSAVE
CALL
CDBL
CHR$
CINT
CIRCLE
CLEAR
CLOAD
CLOSE
CLS
CMD
COLOR
CONT
COPY
COS
CSAVE
CSNG
CSRLIN
CVD
CVI
CVS
DATA
DEF

DEFDBL
DEFINT
DEFSNG
DEFSTR
DELETE
DIM
DRAW
DSKF
DSKI$
DSKO$
ELSE
END
EOF
EQV
ERASE
ERL
ERR
ERROR
EXP
FIELD
FILES
FIX
FN
FOR
FPOS
FRE
GET
GO TO
GOSUB
GOTO
HEX$
IF
IMP

INKEY$
INP
INPUT
INSTR
INT
IPL
KEY
KILL
LEFT$
LEN
LET
LFILES
LINE
LIST
LLIST
LOAD
LOC
LOCATE
LOF
LOG
LPOS
LPRINT
LSET
MAX
MERGE
MID$
MKD$
MKI$
MKS$
MOD
MOTOR
NAME
NEW

NEXT
NOT
OCT$
OFF
ON
OPEN
OR
OUT
PAD
PAINT
PDL
PEEK
PLAY
POINT
POKE
POS
PRESET
PRINT
PSET
PUT
READ
REM
RENUM
RESTORE
RESUME
RETURN
RIGHT$
RND
RSET
RUN
SAVE
SCREEN
SET

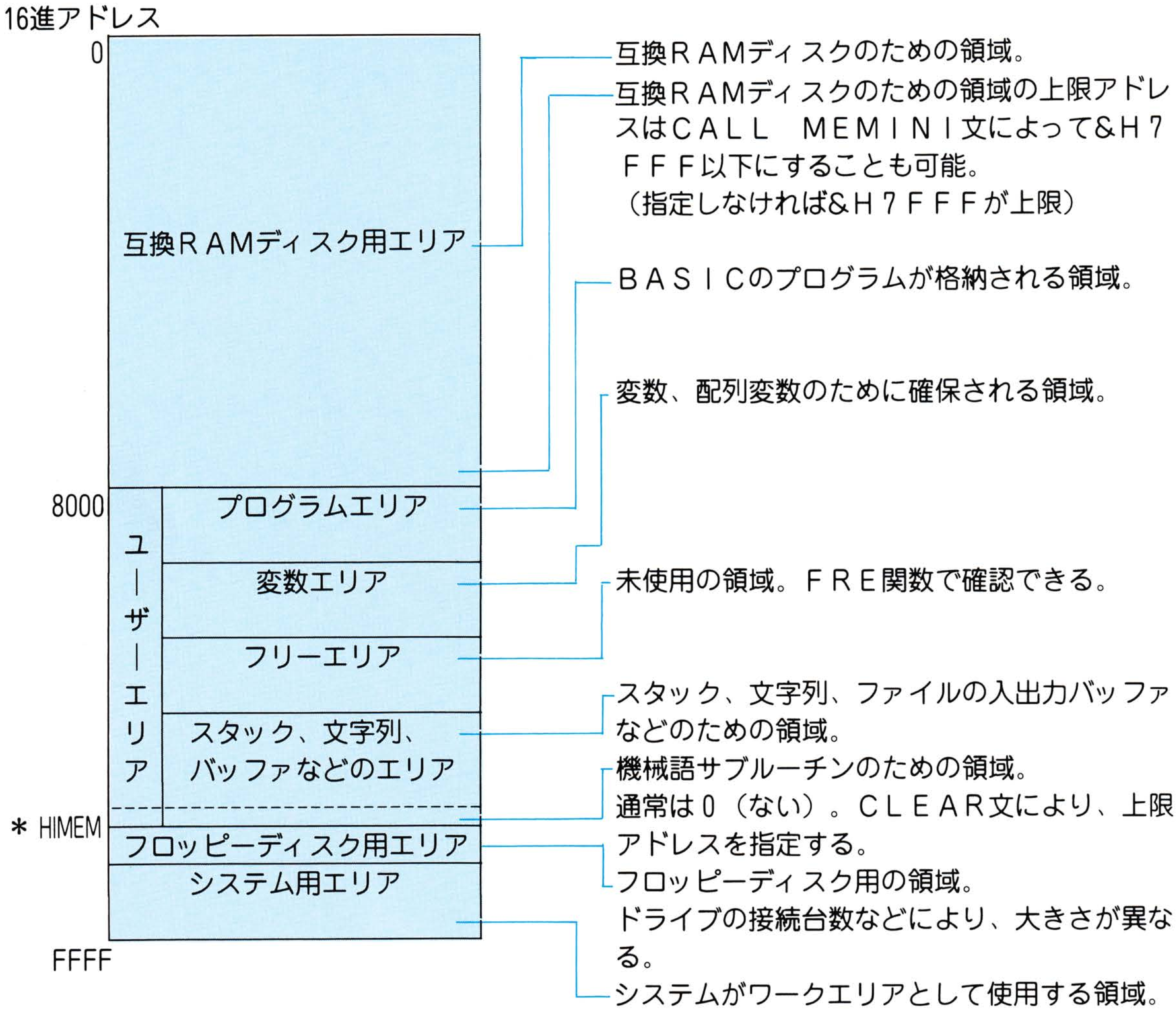
SGN
SIN
SOUND
SPACE$
SPC(
SPRITE
SQR
STEP
STICK
STOP
STR$
STRIG
STRING$
SWAP
TAB(
TAN
THEN
TIME
TO
TROFF
TRON
USING
USR
VAL
VARPTR
VDP
VPEEK
VPOKE
WAIT
WIDTH
XOR

コントロールコード表

コード（10進）	コード（16進）	機　能	対応キー
0	00		
1	01	グラフィックキャラクタの入出力時のヘッダ	CTRL + A
2	02	カーソルを直前の語の先頭に移動	CTRL + B
3	03	入力待ち状態を終了する	CTRL + C
4	04		CTRL + D
5	05	カーソル以下を削除	CTRL + E
6	06	カーソルを次の語の先頭へ移動	CTRL + F
7	07	ＢＥＥＰ音を鳴らす（ＢＥＥＰ文と同じ）	CTRL + G
8	08	カーソルの１つ前の文字を削除する	CTRL + H, BS
9	09	次の水平タブ位置に移動	CTRL + I, TAB
10	0A	カーソルを次の行に送る（ラインフィード）	CTRL + J
11	0B	カーソルをホームポジション（画面左上角）に戻す	CTRL + K
12	0C	画面を消去し、カーソルをホームポジションに戻す	CTRL + L
13	0D	カーソルを画面の左端に戻す（キャリッジリターン）	CTRL + M, ↵
14	0E	カーソルを現在の行の行末に移動	CTRL + N
15	0F		CTRL + O
16	10		CTRL + P
17	11		CTRL + Q
18	12	挿入モードのＯＮ／ＯＦＦスイッチ	CTRL + R, INS
19	13		CTRL + S
20	14		CTRL + T
21	15	現在カーソルがある１行を画面から削除	CTRL + U
22	16		CTRL + V
23	17		CTRL + W
24	18		CTRL + X, SELECT
25	19		CTRL + Y
26	1A		CTRL + Z
27	1B		CTRL + [, ESC
28	1C	カーソルを右へ１文字移動	CTRL + ¥, >
29	1D	カーソルを左へ１文字移動	CTRL +], <
30	1E	カーソルを上へ１文字移動	CTRL + ^, ∧
31	1F	カーソルを下へ１文字移動	CTRL + _, ∨
127	7F	カーソルの位置の１文字を削除	DEL

メモリマップ(BASIC使用時)

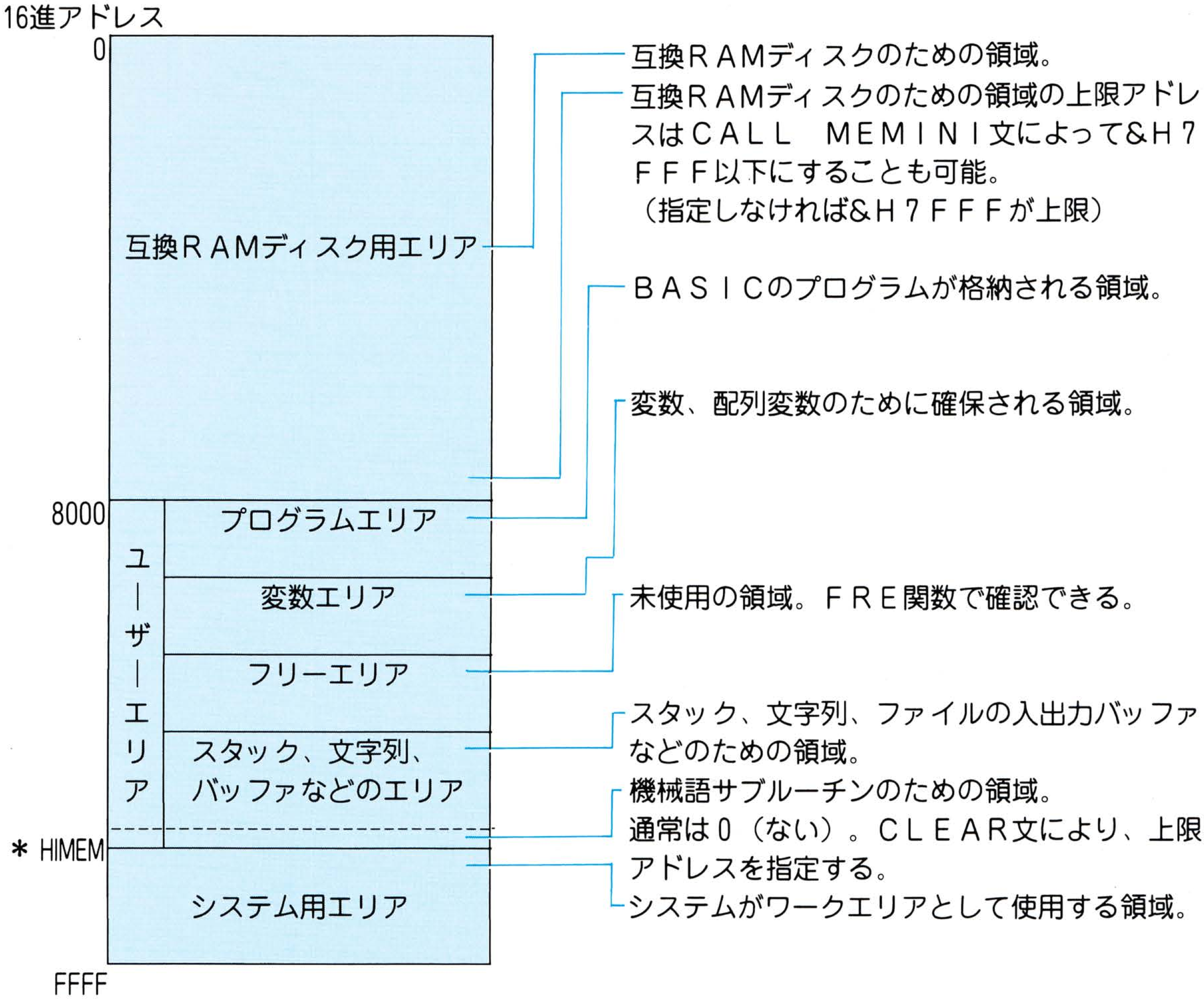
フロッピーディスクドライブを内蔵または接続している場合



＊ＨＩＭＥＭのアドレスは、&ＨＦＣ４Ａと&ＨＦＣ４Ｂ番地に格納されている。

ＨＩＭＥＭ＝（&ＨＦＣ４Ｂ）×&Ｈ１００＋（&ＨＦＣ４Ａ）

シフト立ち上げをした場合



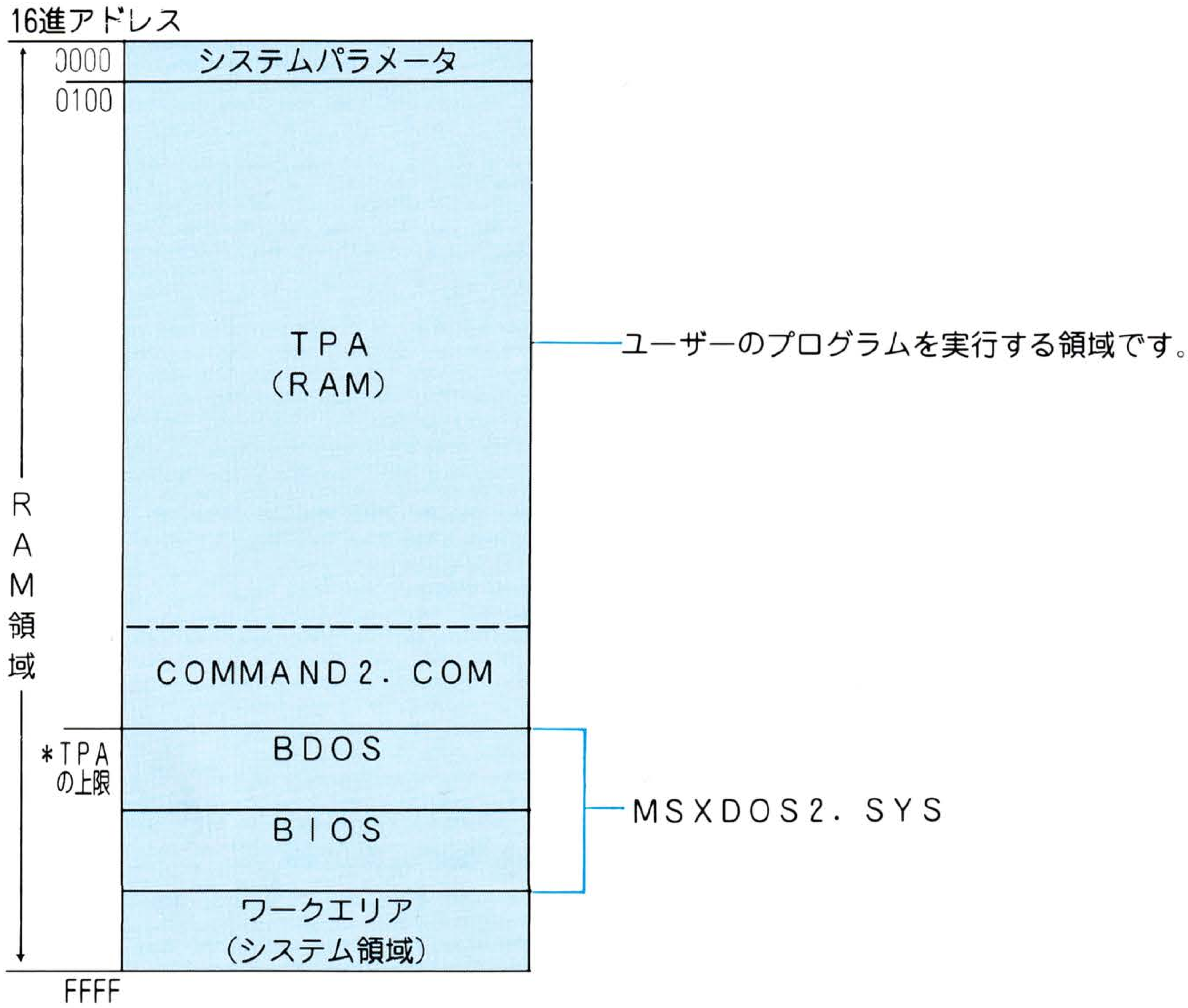
＊ＨＩＭＥＭのアドレスは、&ＨＦＣ４Ａと&ＨＦＣ４Ｂ番地に格納されている。

ＨＩＭＥＭ＝（&ＨＦＣ４Ｂ）×&Ｈ１００＋（&ＨＦＣ４Ａ）

※ドライブを内蔵または接続している場合でも、ＢＡＳＩＣのスタート時に SHIFT を押しながらスタートしたときは、この状態になります。

資料編

メモリマップ(MSX-DOS2使用時)



*TPAの上限は、&H6番地と&H7番地に格納されています。

MSX-DOS2リファレンス表

ＭＳＸ－ＤＯＳ２のファイルスペック

＜ドライブ名＞：＜パスリスト＞＜ファイル名＞＜拡張子＞　　例）Ａ：GAME￥GAMEPROG．BAS
「″」で囲んで使用することはできない。
＜ドライブ名＞：を省略したときには、現在「プロンプト」（Ａ＞など）に表示されているドライブが指定される。
使用できる文字の種類、文字数、ワイルドカード、その他についてはＢＡＳＩＣと同じ。

ＭＳＸ－ＤＯＳ２の区切り記号

命令とそれに続く項目のあいだ、および各項目のあいだには必ず１文字分の空白を入れる。

カレントドライブの切り替え

「コマンドライン」（現在カーソルが表示されているプロンプトに続く行）にドライブ名を入力して ⏎ キーを押すことにより、プロンプトに表示されるドライブ名を変更することができる。

ＭＳＸ－ＤＯＳ２のディスクフォーマット

ＭＳＸ－ＤＯＳ２とＢＡＳＩＣのディスクのフォーマットは共通。したがって、どちらを使ってフォーマットしたディスクでも両方で使用することができる。ただしＭＳＸ－ＤＯＳ２のスタートには、ＭＳＸ－ＤＯＳ２のシステムファイルがコピーされているディスクが必要。

＊印の命令　　実行時には、システムディスク２をドライブＡにセットすることが必要。

MSX-DOS2の命令

書式	機能	文例
ＡＳＳＩＧＮ［〈論理ドライブ名〉 ［〈物理ドライブ名〉］］	論理ドライブを物理ドライブに割り当てる	ＡＳＳＩＧＮ　Ｇ：　Ａ： ＡＳＳＩＧＮ
ＡＴＤＩＲ ＋Ｈ／－Ｈ　［／Ｈ］［／Ｐ］ 複合ファイルスペック	ディレクトリの属性を変え、可視または不可視にする ＋Ｈのとき不可視 －Ｈのとき可視 （ただし、－Ｈは「／Ｈ」なしでは意味がない） ／Ｈは不可視のファイルを対象にするオプション ／Ｐは画面出力を１ページごとに停止するオプション 複合ファイルスペックは、単一のファイルスペック、または複数のファイルスペックを「＋」でつないだもの	ＡＴＤＩＲ　＋Ｈ　WORDS ＡＴＤＩＲ　－Ｈ　WORDS／Ｈ ＡＴＤＩＲ　＋Ｈ　￥ＤＲ１＋￥ＤＲ２
ＡＴＴＲＩＢ ＋Ｒ／－Ｒ／＋Ｈ／－Ｈ　［／Ｈ］［／Ｐ］ 複合ファイルスペック	ファイルの属性を変更し、それらを可視または不可視にする、あるいは読み出し専用または書き込み可能にする ＋Ｈのとき不可視 －Ｈのとき可視 ＋Ｒのとき読み出し専用 －Ｒのとき書き込み可能	ＡＴＴＲＩＢ　＋Ｈ　ＭＩＫＥ ＡＴＴＲＩＢ　－Ｒ　－Ｈ　ＭＩＫＥ／Ｈ ＡＴＴＲＩＢ　＋Ｒ　￥ＤＲ１＋ＦＬＳ１
ＢＡＳＩＣ　［〈プログラム名〉］	ＢＡＳＩＣに制御を移す 〈プログラム〉がディスクに保存されていれば、ロードした後、実行される	ＢＡＳＩＣ ＢＡＳＩＣ　″ＢＵＮＮＹ″

書　式	機　能	文　例
BUFFERS　[〈数値〉]	システムが使用するディスクバッファの数を変更する 〈数値〉が省略されたときは現在の数を表示する	BUFFERS BUFFERS　10
CHDIR 　[〈ドライブ名〉]　[〈パスリスト〉] CD 　[〈ドライブ名〉]　[〈パスリスト〉]	カレントディレクトリを変更する 〈パスリスト〉が省略されたときは、カレントディレクトリまでのパスリストを表示	CHDIR　¥WORDS CD
CHKDSK　[〈ドライブ名〉]　[／F] ＊	ファイルシステムの整合性を検査する ／Fを指定すると、発見したエラーを修復する 指定しないと、修復しないでメッセージだけ表示する	CHKDSK CHKDSK　B:／F
CLS	BASICのCLSと同じ	CLS
COMMAND2　[〈コマンド〉]	DOS2のコマンドインタプリタを外部コマンドとして起動する	COMMAND2 COMMAND2　MIKE.BAT
CONCAT [／H]　[／P]　[／B]　[／V]　[／A] 複合ファイルスペック　ファイルスペック	ファイルを連結する 複合ファイルスペックで連結するファイルを指定し、新しく単一のファイルを作る ／Bはバイナリモード 通常は各ファイルのEOFを除いて連結し、最後にEOFを付加するが、そのまま何も手を加えないモード ファイル単位でも指定可能 ／Vは書き込みチェック ／AはASCIIモードへの復帰	CONCAT　＊.TXT　TS.PRN CONCAT　／B　GRP1+GRP2 　+GRP3　MASS.GRP CONCAT　GRP4／B 　+TEXT.DOC／A　FILE1
COPY　[／A]　[／H]　[／T] 　[／V]　[／P]　[／B] 〈ソースファイル〉　〈目的ファイル〉	ファイルまたはデバイスから他のファイルまたはデバイスにデータをコピーする 〈ソースファイル〉は、コピーされる側の複合ファイルスペック 〈目的ファイル〉は、新しく作られる側のファイルスペックまたはデバイス名 ／Tは新しくできたファイルに現在の日付と時刻を記録	COPY　MIKE　PET¥TAMA COPY　MIKE　B: COPY　CON　PRN COPY　＊.SRC　＊.OBJ COPY　XX.LST+YY.LST 　B:ZZZZ.LST／T COPY　＊.＊　¥BACKUP
DATE　[〈日付〉]	現在の日付を表示・設定する 〈日付〉を省略すると、現在の日付を表示し、新しい日付の入力を求める	DATE DATE　95-9-30
DEL	（ERASEと同じ）	

書式	機能	文例
DIR　[／H]　[／W]　[／P] [複合ファイルスペック]	ディスク上のファイル名や ディレクトリ名を表示する ／Wはワイド形式での表示	DIR DIR　／W DIR　¥WORDS¥*．TXT
DISKCOPY　[〈複写元ドライブ名〉 [〈複写先ドライブ名〉]]　[／X] [／S] *	ディスクを別のディスクにコピーする 〈複写元〉と〈複写先〉は、 同じフォーマットのディスクであることが必要 ／Xはメッセージ表示の省略 ／Sはシステムディスク作成	DISKCOPY　A：　B： DISKCOPY　B：／X DISKCOPY　B：／S
ECHO　[〈文字列〉]	〈文字列〉を画面に表示する 省略は改行	ECHO　Who　are　you? ECHO
ERA	（ERASEと同じ）	
ERASE　[／H]　[／P] 複合ファイルスペック ERA　[／H]　[／P] 複合ファイルスペック DEL　[／H]　[／P] 複合ファイルスペック	１つまたは複数のファイルを削除する 属性が読み出し専用のファイルは削除できない	ERASE　POCHI．BAT ERA　WORDS¥FILE?．DOC DEL　*．COM／H ERASE　¥PET
EXIT　[〈エラーコード〉]	現在のCOMMAND2．COMを終了する 外部コマンドとして呼び出したコマンドインタプリタを終了するコマンド 終了後は呼び出したプログラムに戻る 〈エラーコード〉は呼び出したプログラムに渡す数値	EXIT EXIT　40
FIXDISK [〈ドライブ名〉]　[／S] *	MSX－DOSでフォーマットされたディスクをMSX－DOS２のフォーマットに変更する ／Sはディスクのブートセクタを書き換え、UNDELコマンドを使用可能にする ／Sを付けて実行すると、MSX－DOS上で動作する市販ソフトなどの一部は、動作しなくなることがある	FIXDISK FIXDISK　／S
FORMAT　[〈ドライブ名〉]	ディスクをフォーマット（初期化）する ディスクに保存されている内容はすべて消える	FORMAT
HELP　[〈項目〉] *	MSX－DOS２のコマンドや主な機能についての解説を表示する 解説は、システムディスクの「KHELP」というサブディレクトリにある「．HLP」という拡張子をもつファイルの表示により行われる	HELP HELP　FORMAT

書式	機能	文例
KMODE {〈番号〉 / OFF} KMODE {[〈番号〉] / [OFF]} ／S [〈ドライブ名〉] *	漢字モードと英数・かなモードを切り替える 〈番号〉は漢字モードの番号0～3（⇨漢字BASIC） OFFで英数・かなモード ／Sは次回からそのモードでスタートするようにディスクのブートセクタを書き換える	KMODE KMODE 3 KMODE ／S
MD	（MKDIRと同じ）	
MKDIR [〈ドライブ名〉] [〈パスリスト〉¥]〈新ディレクトリ名〉 MD [〈ドライブ名〉] [〈パスリスト〉¥]〈新ディレクトリ名〉	新しいサブディレクトリを作る 〈パスリスト〉は、カレントディレクトリからサブディレクトリを新しく作る位置までのパスリストに新ディレクトリ名を「¥」でつないだもの 新ディレクトリ名だけの場合はカレントディレクトリに作られる	MKDIR DOG MD HOBBY¥PET¥DOG
MODE 〈桁数〉	画面の1行の文字数を変更する 〈桁数〉は1～80	MODE 80 MODE＝25
MOVE [／H] [／P] 複合ファイルスペック [〈パスリスト〉]	ファイルを同じディスクの別のディレクトリに移す	MOVE DOC1 +DOC2 WORDS¥TEXT MOVE ／H *.COM ¥
MVDIR [／H] [／P] 複合ファイルスペック [〈パスリスト〉]	サブディレクトリを同じディスクの別のディレクトリに移す 複合ファイルスペックは移動するディレクトリ名で、省略はカレントディレクトリ	MVDIR WORDS¥TEXT LTRS MVDIR ／H * ¥
PATH [{[+] / [−]} [〈ドライブ名〉]〈パスリスト〉 [[〈ドライブ名〉]〈パスリスト〉 ……]]]	COMおよびBATコマンド検索用パスリストを表示・設定する すべての指定を省略すると、現在設定されている検索用パスリストを表示する ーは、現在のパスリストから入力したパスリストと同じ部分を削除する ＋は、現在のパスリストから入力したパスリストと同じ部分を削除し、末尾に入力したパスリストを加える	PATH PATH +¥MYBAT PATH −A:¥MYBAT

書　式	機　能	文　例
ＰＡＵＳＥ　[〈文字列〉]	バッチファイル中でプロンプトを表示しキー入力を待つ バッチファイル実行中の操作指示などに使う 〈文字列〉はプロンプトの前に表示される プロンプトはキー入力を求めるメッセージ	ＰＡＵＳＥ ＰＡＵＳＥ　Ａｒｅ　ｙｏｕ　ｓｕｒｅ？
ＲＡＭＤＩＳＫ [〈容量〉[Ｋ]]　[／Ｄ]	ＲＡＭディスクの大きさを設定・表示する すべての指定を省略すると現在設定されているＲＡＭディスクの大きさをキロバイト単位で表示する 〈容量〉はキロバイト単位で０～４０６４の数字で入力し１６キロバイト単位の端数切り上げで設定される ／ＤはＲＡＭディスクを削除する（〈容量〉０も同じ）	ＲＡＭＤＩＳＫ ＲＡＭＤＩＳＫ　６４ ＲＡＭＤＩＳＫ＝１２８ ＲＡＭＤＩＳＫ　／Ｄ
ＲＤ	（ＲＭＤＩＲと同じ）	
ＲＥＭ　[〈文字列〉]	バッチファイル中にコメントを入れる バッチファイルの実行には影響しない（メモとして使用）	ＲＥＭ ＲＥＭ　Ｔｈｉｓ　ｉｓ　ｍｙｐｒｏｇ．
ＲＥＮ	（ＲＥＮＡＭＥと同じ）	
ＲＥＮＡＭＥ　[／Ｈ]　[／Ｐ] 複合ファイルスペック　ファイル名 ＲＥＮ　[／Ｈ]　[／Ｐ] 複合ファイルスペック　ファイル名	１つまたは複数のファイル名を変更する 複合ファイルスペックは名前を変更するファイル ファイル名は新しい名前 一度に複数のファイルの名前を変更する場合、新ファイル名の一部にワイルドカードを使用して元の名前の一部を残すことが可能	ＲＥＮＡＭＥ ＵＴＩＬＳ￥ＭＹＢＡＴ　ＭＹＣＯＭ ＲＥＮ　ＣＯＭ１＋ＣＯＭ２　ＭＹＰ？ ＲＥＮＡＭＥ　＊．ＤＯＣ　＊．ＴＸＴ ＲＥＮ　ＴＡＲＯ．＊　ＪＩＲＯ．＊
ＲＭＤＩＲ [／Ｈ]　[／Ｐ]　複合ファイルスペック ＲＤ [／Ｈ]　[／Ｐ]　複合ファイルスペック	１つまたは複数の、空のサブディレクトリを削除する 複合ファイルスペックで削除するサブディレクトリを指定 ファイルやサブディレクトリがあるディレクトリは削除できない	ＲＭＤＩＲ　ＥＡＴ ＲＭＤＩＲ　ＨＯＢＢＹ￥ＰＥＴ ＲＤ　ＤＲＩＮＫ＋ＧＡＭＥ＋ＢＵＹ
ＲＮＤＩＲ　[／Ｈ]　[／Ｐ] 複合ファイルスペック　ディレクトリ名	１つまたは複数のサブディレクトリの名前を変更する 複合ファイルスペックで名前を変更するサブディレクトリを指定 ディレクトリ名は新しい名前	ＲＮＤＩＲ　ＭＯＶＹ　ＴＡＬＫＹ ＲＮＤＩＲ　ＨＯＢＢＹ￥ＰＥＴ　ＺＯＯ ＲＮＤＩＲ　ＨＯＢＢＹ￥＊　＊．ＨＯＢ

資料編

書　式	機　能	文　例
ＳＥＴ ［〈名前〉］［〈区切り記号〉］［〈値〉］	環境変数を設定・表示する すべての指定を省略すると、現在定義されているすべての環境変数と現在の値を表示 〈名前〉は環境変数の変数名 〈名前〉だけが指定されるとその環境変数の現在の値が表示される 〈区切り記号〉は、空白、「＝」、「;」、「，」などで、〈値〉を変更するときは省略できない 〈値〉は、文字列または数値で環境変数により異なる 〈区切り記号〉を省略しないで〈値〉だけを省略すると、その環境変数は削除される	ＳＥＴ ＳＥＴ　ＨＥＬＰ ＳＥＴ　ＴＩＭＥ＝１２
ＴＩＭＥ　［〈時刻〉］	現在の時刻を設定・表示する 〈時刻〉を省略すると、設定されている時刻を表示し、新しい時刻の入力を求める 時刻は４組の数字で入力し、次のいずれかの記号で区切る 空白　タブ　，　－　．　： 省略した組の数字は、現在の値のままになる	ＴＩＭＥ ＴＩＭＥ　９５－９－３０－００ ＴＩＭＥ　，，２９，
ＴＹＰＥ　［／Ｈ］［／Ｐ］［／Ｂ］ ｜複合ファイルスペック｜ ｜デバイス名｜	ファイルの内容またはデバイスからのデータを表示する ファイルの内容を表示するときは複合ファイルスペック デバイスからの入力データを表示するときはデバイス名 ／Ｂを指定するとＡＳＣＩＩデータ以外でも表示する	ＴＹＰＥ　ＲＥＢＯＯＴ．ＢＡＴ ＴＹＰＥ ＮＥＷＳ￥＊．ＮＯＷ＋￥ＢＯＸ ＴＹＰＥ　ＣＯＮ
ＵＮＤＥＬ　［ファイルスペック］ ＊	前に削除されたファイルを復活する ファイルスペックは復活したいファイルの指定 省略時は「＊．＊」 復活できるのは、ＭＳＸ－ＤＯＳ２フォーマットのディスクでＭＳＸ－ＤＯＳ２を使って削除したファイルで、削除後書き込みしていないもの 削除したディレクトリも復活	ＵＮＤＥＬ　ＰＯＩ．ＤＯＣ ＵＮＤＥＬ　ＬＴＲＳ￥＊．ＴＸＴ ＵＮＤＥＬ
ＶＥＲ	動作中のシステムのバージョン番号を表示する	ＶＥＲ

書式	機能	文例
VERIFY [ON] | [OFF]	ディスクの書き込みチェックを設定・解除する 指定を省略すると、現在の書き込みチェックモードを表示 ONで設定 OFFで解除 書き込みチェックを設定すると、書き込み時の信頼性が向上する代わりに時間がかかる	VERIFY VERIFY ON VERIFY OFF
VOL [〈ドライブ名〉] [〈ボリューム名〉]	ディスクのボリューム名を表示・変更する ディスク1枚ごとにつける名前がボリューム名 最大、半角11文字または全角5文字まで 省略すると、現在のボリューム名を表示	VOL VOL BACKUP
XCOPY [ファイルスペック[ファイルスペック]] [オプション] *	ディスクのファイルやディレクトリを別のディスクにコピーする 最初のファイルスペックは複写元のファイル名またはディレクトリ名 2番目のファイルスペックは新しく作るファイル名またはディレクトリ名 使用できるオプションは次のとおり ／A ／E ／H ／M ／P ／S ／T ／W ／V ／Tはコピーされたファイルの日付と時刻を現在の日付と時刻にする ／Aはアーカイブ属性がセットされているファイルだけをコピーする ／Mはファイルをコピーしたあとで、アーカイブ属性をリセットする ／Sはファイルをディレクトリごとコピーし、サブディレクトリやそのファイルもコピーする（空のディレクトリはコピーしない） ／Eは／Sが指定されているときに空のディレクトリもコピーする ／Pは1ファイルごとにコピーするかどうか確認する	XCOPY A:¥ B: XCOPY *.* B: ／H／S／M

書　式	機　能	文　例
XCOPY（つづき）	／Wはコピーを始める前にプロンプトを表示するのでディスクを入れ替えできる ／Vは書き込みチェック指定	
XDIR　［ファイルスペック］［／H］ *	ディレクトリの下のすべてのファイルの一覧を表示する ファイルスペックは表示するファイルの指定 指定のディレクトリのファイル名を表示すると、サブディレクトリのファイル名を字下げして表示する	XDIR XDIR　B：HOBBY¥PET XDIR　¥*．COM／H

BASICエラーメッセージ表

※💾はドライブを内蔵または接続している場合のみ表示

メッセージ	エラーコード	説　明
Bad drive name 💾	6 2	ドライブ名の指定がまちがっている
Bad FAT 💾	6 0	ディスクが正しくフォーマットされていない。またはこわれている。
Bad file mode 💾	6 1	シーケンシャルファイルに対してＰＵＴ文、ＧＥＴ文またはＬＯＦ関数が使用されたり、ランダムファイルに対してＬＯＡＤ文が使用された。またはＯＰＥＮ文でモードの設定をまちがえている。
Bad file name	5 6	ファイル名が適切でない。
Bad file number	5 2	ＯＰＥＮしていない（開いていない）ファイル番号を指定した。指定したファイル番号がＭＡＸＦＩＬＥＳ文で指定した数を越えている。
Can´t CONTINUE	1 7	ＣＯＮＴ命令でプログラムの実行を再開できない。 エラーが発生して終了したときは再開できません。また、プログラムの一部を書き換えたり、ＣＬＥＡＲ文などを実行すると、次に実行する文を示すポインタが壊れて再開できなくなります。
Device I/O error	1 9	プリンタなどへの入出力中にエラーが起こった。
Direct statement in file 💾	5 7	アスキー形式のファイルをロード中、ファイルの中に命令のコードが直接入っていた。
Directory already exists 💾	7 3	すでに同じ名前のディレクトリがある。
Directory not found 💾	7 4	指定されたディレクトリが見つからない。
Disk full 💾	6 6	ディスク内に空き領域が無い。
Disk I/O error 💾	6 9	ディスク入出力操作において、回復できないエラーが発生した。
Disk offline 💾	7 0	指定されたドライブにディスクが入っていない。 またはディスクがフォーマットされていない。
Disk write protected 💾	6 8	ディスクのライトプロテクトタブが書き込み禁止の位置にあるのに、書き込みをともなう操作を行おうとした。
Division by zero	1 1	０による除算を行おうとした。または０に対して負の累乗を行おうとした。
FIELD overflow 💾	5 0	ＦＩＥＬＤ文で定義したデータの長さの合計が、１レコードの大きさを越えた。（レコード長未定義のときは合計256バイト以内）
File already exists	6 5	ＮＡＭＥ文などで指定された新しいファイル名のファイルが、すでに存在している。
File already open	5 4	すでに開いているファイルを再びＯＰＥＮしようとした。または、同じファイル番号で開いたファイルをまだＣＬＯＳＥしていない。 あるいは、開いているファイルをＫＩＬＬしようとした。
File not found 💾	5 3	指定されたファイルが見つからない。
File not OPEN	5 9	ＯＰＥＮ文で開いていないファイルに対して入出力を行おうとした。
File still open	6 4	ファイルのＣＬＯＳＥを忘れている。
File write protected 💾	7 2	ファイルが読み出し専用になっている。
Illegal direct	1 2	ダイレクトモードでは使用できない命令をダイレクトモードで使おうとした。
Illegal function call	5	命令や関数の使いかた（数値の範囲や省略のしかたなど）がまちがっている。
Input past end	5 5	ファイルのデータをすべて読み込んだあと、さらにデータを入力しようとした。（ファイルのデータをすべて読み込んだかどうかは、ＥＯＦ関数で調べてください。）
Internal error	5 1	ＢＡＳＩＣ内部でエラーが発生した。 （電源スイッチを入れ直してみてください）

メッセージ	エラーコード	説　　明
Line buffer overflow	25	入力されたプログラムやデータが許された文字数を越えている。
Missing operand	24	入力された文の必要な項目が省略されている。
NEXT without FOR	1	ＮＥＸＴ文に対応するＦＯＲ文がない。
No RESUME	21	エラー処理ルーチンにＲＥＳＵＭＥ文がない。
Out of DATA	4	ＲＥＡＤ文で読み出すデータが不足している。
Out of memory	7	プログラムが長すぎたり、配列が大きすぎたりしてメモリが足りなくなった。
Out of string space	14	メモリの文字型変数用の領域が足りなくなった。 （文字列領域の大きさは、ＣＬＥＡＲ文である程度調節可能）
Overflow	6	計算の結果や入力された数値が、型に応じた数値の範囲を越えた。
RAM disk already exists	75	ＲＡＭディスクの容量を０にする前に、ＲＡＭディスクの容量を設定し直そうとした。
Redimensioned array	10	すでに定義した配列変数を再び定義しようとした。または定義せずに使用した配列変数を、あとから定義しようとした。
Rename across disk	71	異なるドライブのあいだでＲＥＮＡＭＥ文を実行させようとした。
RESUME without error	22	エラーが発生していないのにＲＥＳＵＭＥ文を実行させようとした。 またはエラーなしでエラー処理ルーチンを使おうとした。
RETURN without GOSUB	3	ＲＥＴＵＲＮ文に対応するＧＯＳＵＢ文がない。
Sequential I/O only	58	シーケンシャルファイルに対してランダム入出力を行おうとした。
String formula too complex	16	指定した文字列が複雑すぎる。 （文字列を２つ以上に分けてください。）
String too long	15	文字列が 255文字より長い。
Subscript out of range	9	配列変数の添え字の値が、ＤＩＭ文で定義した最大値を越えた。
Syntax error	2	文の書きかた（構文）をまちがえている。（かっこが正しく対応していない。命令や項目の綴りがちがう。区切り記号がないなど）
Too many files	67	指定されたディスク内のファイルの数が 112個に達しているときに、新しいファイルを作ろうとした。（１ディスクは 112ファイルまで）
Type mismatch	13	数式や数値型変数に文字列を使ったり、文字式や文字型変数に数値を使うなど、データの形式が合っていない。
Undefined line number	8	ＧＯＴＯ文、ＧＯＳＵＢ文、ＩＦ文などで指定した行番号の行が、プログラム中に存在しない。
Undefined user function	18	ＤＥＦ　ＦＮ文で定義されていないユーザー定義サブルーチンを使おうとした。
Unprintable error	23 26~49 76~255	メッセージが定義されていないエラーが発生した。
Verify error	20	ディスクの書き込みチェック（VERIFY）がオンのときのみ表示。 ディスクに正しく書き込めていない。 （ディスクをセットし直し、同じ操作をして見てください。）

MSX-DOS2エラーメッセージ表

日本語メッセージ	英文メッセージ	説　　明
．や．．に対しては操作できません	Invalid . or .. operation	「．」（カレントディレクトリ）や「．．」（すぐ上のディレクトリ）に対して使用できない命令である。
ＭＳＸ－ＤＯＳディスクではありません	Not a DOS disk	ＭＳＸ－ＤＯＳ２が読み出せるフォーマットではないディスクを読み書きしようとした。
書き込み異常です	Write error	ディスクへの書き込みに失敗した。
このディスクは使用できません	Incompatible disk	２Ｄや１Ｄのディスクを読み書きしようとした。
このドライブはフォーマットできません	Cannnot format this drive	ＲＡＭディスクをフォーマットしようとした。ＲＡＭディスクはフォーマットできません。
コマンドが違います	Unrecognized command	命令の入力まちがい。または、入力された名前の外部コマンドやバッチファイルが見つからない。
コマンドが長すぎます	Command too long	バッチファイル中に入力されている命令の１行の文字数が多すぎる。
コマンドのバージョンが違います	Wrong version of command	プログラムのスタート時とは異なるバージョンのシステムディスクがセットされている。
シークエラーです	Seek error	指定されたトラックがディスクに見つからない。
自分自身にはコピーできません	File cannot be copied onto itself	ＣＯＰＹ命令などで、ソースファイルと目的ファイルが同じ名前で同じディレクトリにある。
セクターが見つかりません	Sector not found	指定されたセクタがディスクに見つからない。
正しく書き込まれませんでした	Verify error	書き込みチェックモードのとき、チェックしたデータが正しく書き込まれていなかった。
ディスクが異常です	Disk error	ディスクが異常で、データを読み書きできない。
ディスクがいっぱいです	Disk full	ディスクの空き容量が不足している。ディスクの交換か、不要ファイルの削除が必要です。
ディスクが書き込み保護されています	Write protected disk	ディスクのライトプロテクトタブが書き込み保護の位置にセットされているのに、ディスクへの書き込みを行う操作をしようとした。
ディスクが違います	Wrong disk	正しいディスクをセットし、Ｒキー（「Retry」）を押す。
ディスクが入っていません	Not ready	指定のドライブにディスクがセットされていない。

日本語メッセージ	英文メッセージ	説　　明
ディスクが フォーマット されていません	Unformatted disk	指定したドライブのディスクは、まだフォーマットされていないので使用できない。フォーマットしてから使用してください。
ディスクの データが異常です	Data error	読み込んだデータに異常がある。
ディレクトリが 移動できません	Invalid directory move	MVDIR命令で、ディレクトリをそれ自身のサブディレクトリに移動しようとした。
ディレクトリが 空ではありません	Directory not empty	RMDIR命令で、ファイルやサブディレクトリがあるディレクトリを削除しようとした。
ディレクトリが 既にあります	Directory exists	すでにあるディレクトリと同じ名前のディレクトリを作ろうとした。
ディレクトリが 見つかりません	Directory not found	指定されたディレクトリが見つからない。
パス名が長過ぎます	Pathname too long	パスリストの文字数が63文字を越えている。または、指定したファイルが、ルートディレクトリから63文字以上のパスリストが必要な位置にある。
パラメータが 多すぎます	Too many parameters	命令に必要な項目数以上の項目を入力した。
パラメータが 不足しています	Missing parameter	その命令に必要な項目が入力されていない。
ファイルが 使用中です	File is already in use	バッチファイル実行中などに、別の目的で使用中のファイルを変更しようとした。
ファイルが 既にあります	File exists	MKDIR命令で、すでにあるファイル名またはディレクトリ名を新ディレクトリ名に指定した。
ファイルが 見つかりません	File not found	指定されたファイルが見つからない。
ファイルが 読み出し専用です	Read only file	読み出し専用のファイルを変更または上書きしようとした。
ファイルの 重ね書きが できません	Cannot overwrite previous destination file	COPY命令でワイルドカードにより目的ファイルのファイル名を複数指定したとき、複数の目的ファイルが同じファイル名になる。
ファイル名が 重複しています	Duplicate filename	RENAME等で、すでにあるファイル名またはディレクトリ名を変更後の名前として指定した。

日本語メッセージ	英文メッセージ	説　　明
ファイルを作成できません	Cannot create destination file	ＣＯＰＹ命令で、目的ファイルのファイル名が現在使用中のファイルと同じである。
ＦＡＴ異常です	Bad file allocation table	ディスクが壊れている。
複写先ファイルは結合できません	Cannot concatenate destination file	ＣＯＮＣＡＴ命令で、ソースファイルのひとつとして、結合後の目的ファイルを指定した。
不正なファイル名です	Invalid filename	ファイル名に不正な文字がある。
ＨＥＬＰファイルが見つかりません	File for ＨＥＬＰ not found	ＨＥＬＰ命令で、指定された命令のＨ･ＥＬＰファイル（説明文）が見つからない。
無効なオプション指定です	Invalid option	命令の中で指定されたオプションがまちがっている。
無効な環境変数です	Invalid environment string	環境変数の前に不正な文字がある。
無効な時間です	Invalid time	入力された時刻がまちがっている。
無効な数値です	Invalid number	数値を入力する項目に文字を入力した。
無効な属性です	Invalid attributes	ＡＴＴＲＩＢ命令またはＡＴＤＩＲ命令で、不正な属性が指定された。
無効なデバイスオペレーションです	Invalid device operation	システムが使用するデバイス名に対して、使用できない命令である。
無効なドライブ名です	Invalid drive	ドライブ名の指定がまちがっている。
無効なパス名です	Invalid pathname	指定されたパスリストが見つからない。
無効なパラメータです	Invalid parameter	入力された命令文の項目がまちがっている。
無効な日付けです	Invalid date	入力された日付がまちがっている。
メモリー不足です	Not enough memory	指定された命令を実行できるだけのメモリがない。
メディアの種類が違います。	The two disks are incompatible with each other	規格の異なるディスク（２ＤＤと１ＤＤ、フロッピーディスクとＲＡＭディスク）の間で、ＤＩＳＫＣＯＰＹ命令を使おうとした。
ＲＡＭ　ＤＩＳＫがありません	RAM disk does not exist	まだ設定していないＲＡＭディスクの容量を表示させようとした。
ルートディレクトリがいっぱいです	Root directory full	ルートディレクトリにすでに 112個のファイルがあり、ルートディレクトリにはもう増やせない。

ローマ字・かな変換表(BASIC使用時)

ＢＡＳＩＣの英数・かなモードで SHIFT を押しながら かな を押すと、ローマ字・かな変換でひらがなまたはカタカナを入力することができます。（ひらがな・カタカナは CAPS で切り替えます）

50音の対照

あ	A	い	I	う	U	え	E	お	O
か	KA	き	KI	く	KU	け	KE	こ	KO
さ	SA	し	SI SHI	す	SU	せ	SE	そ	SO
た	TA	ち	TI CHI	つ	TU TSU	て	TE	と	TO
な	NA	に	NI	ぬ	NU	ね	NE	の	NO
は	HA	ひ	HI	ふ	HU FU	へ	HE	ほ	HO
ま	MA	み	MI	む	MU	め	ME	も	MO
や	YA			ゆ	YU			よ	YO
ら	RA	り	RU	る	RU	れ	RE	ろ	RO
わ	WA			う	WU			を	WO
ん	NN								

濁音・半濁音の対照

が	GA	ぎ	GI	ぐ	GU	げ	GE	ご	GO
ざ	ZA	じ	ZI	ず	ZU	ぜ	ZE	ぞ	ZO
			JI						
だ	DA	ぢ	DI	づ	DU	で	DE	ど	DO
ば	BA	び	BI	ぶ	BU	べ	BE	ぼ	BO
ぱ	PA	ぴ	PI	ぷ	PU	ぺ	PE	ぽ	PO

拗音・促音の対照

		うぃ	WI			うぇ	WE		
きゃ	KYA	きぃ	KYI	きゅ	KYU	きぇ	KYE	きょ	KYO
ぎゃ	GYA	ぎぃ	GYI	ぎゅ	GYU	ぎぇ	GYE	ぎょ	GYO
しゃ	SYA	しぃ	SYI	しゅ	SYU	しぇ	SYE	しょ	SYO
	SHA				SHU		SHE		SHO
じゃ	ZYA	じぃ	ZYI	じゅ	ZYU	じぇ	ZYE	じょ	ZYO
	JYA		JYI		JYU		JYE		JYO
	JA				JU		JE		JO
ちゃ	TYA	ちぃ	TYI	ちゅ	TYU	ちぇ	TYE	ちょ	TYO
	CYA		CYI		CYU		CYE		CYO
	CHA				CHU		CHE		CHO
ぢゃ	DYA	ぢぃ	DYI	ぢゅ	DYU	ぢぇ	DYE	ぢょ	DYO
てゃ	THA	てぃ	THI	てゅ	TYU	てぇ	TYE	てょ	TYO
でゃ	DHA	でぃ	DHI	でゅ	DYU	でぇ	DYE	でょ	DYO
にゃ	NYA	にぃ	NYI	にゅ	NYU	にぇ	NYE	にょ	NYO
ひゃ	HYA	ひぃ	HYI	ひゅ	HYU	ひぇ	HYE	ひょ	HYO
びゃ	BYA	びぃ	BYI	びゅ	BYU	びぇ	BYE	びょ	BYO
ぴゃ	PYA	ぴぃ	PYI	ぴゅ	PYU	ぴぇ	PYE	ぴょ	PYO
ふぁ	FA	ふぃ	FI			ふぇ	FE	ふぉ	FO
ふゃ	FYA	ふぃ	FYI	ふゅ	FYU	ふぇ	FYE	ふょ	FYO
みゃ	MYA	みぃ	MYI	みゅ	MYU	みぇ	MYE	みょ	MYO
りゃ	RYA	りぃ	RYI	りゅ	RYU	りぇ	RYE	りょ	RYO
ゔぁ	VA	ゔぃ	VI	ゔ	VU	ゔぇ	VE	ゔぉ	VO

小文字の対照

ぁ	LA	ぃ	LI	ぅ	LU	ぇ	LE	ぉ	LO
ゃ	LYA	ぃ	LYI	ゅ	LYU	ぇ	LYE	ょ	LYO

※「っ」は同じ子音を2回押します。
※「X」を押すと「ー」（長音）が表示されます。
※この表は、英数・かなモード時に変換される文字をあらわしています。漢字モードでも、半角文字の入力時は同じです。
※漢字モードの全角文字入力時でのローマ字・かな変換は、次の点が異なります。
・「゛」と「゜」は前の文字と合わせて1文字になります。　例）「が」、「ぱ」
・全角文字には「ヴ」はありますが、「う゛」がありません。「う」と「゛」の2文字で入力してください。
・記号は次の通りになります。

″／″（スラッシュ）……………………″・″（中点）
″.″（ピリオド）……………………″。″（句点）
″,″（カンマ）……………………″、″（読点）
SHIFT ＋″.″……………………″.″（ピリオド）
SHIFT ＋″,″……………………″,″（カンマ）
[（左角かっこ）……………………″「″（左かぎかっこ）
]（右角かっこ）……………………″」″（右かぎかっこ）

資料編

命令索引

BASICとMSX－DOS２の命令、関数、システム変数などの索引です。

A

A： ……37,181
ABS ……188
AND ……52,193
ASC ……190
ASSIGN ……199
ATDIR ……199
ATN ……188
ATTRIB ……199
AUTO ……176

B

B： ……166,167
BASE ……193
BASIC ……142
BEEP ……184
BIN$ ……190
BLOAD ……168,180
BSAVE ……168,180
BUFFERS ……200

C

C： ……166,167
CALL AKCNV ……189
CALL ANK ……99,176
CALL BGM ……184
CALL CHDIR ……165,182
CALL CHDRV ……182
CALL CLS ……115,178
CALL FORMAT ……36,182
CALL JIS ……189
CALL KACNV ……189
CALL KANJI ……111,176
CALL KEXT ……189
CALL KINSTR ……188,190
CALL KLEN ……188,190
CALL KMID ……189
CALL KNJ ……189
CALL KTYPE ……188,190
CALL MEMINI ……176,182
CALL MFILES ……182
CALL MKDIR ……165,182
CALL MKILL ……182
CALL MNAME ……182
CALL MUSIC ……89,184
CALL PALETTE ……178
CALL PAUSE ……177
CALL PCMPLAY ……96,184
CALL PCMREC ……95,184
CALL PITCH ……184
CALL PLAY ……184
CALL RAMDISK ……171,176,182
CALL RMDIR ……165,182
CALL SJIS ……189
CALL STOPM ……184
CALL TEMPER ……184
CALL TRANSPOSE ……184
CALL VOICE ……90,185
CALL VOICE COPY ……185
CD ……200
CDBL ……188
CHDIR ……200
CHKDSK ……200
CHR$ ……55,190
CINT ……188
CIRCLE ……67,178
CLEAR ……176
CLOSE ……168,180
CLS ……16,178
COLOR ……69,178
COLOR SPRITE ……178
COLOR SPRITE$ ……178
COMMAND2 ……200
CONCAT ……200
CONT ……177
COPY ……41,182,200
COS ……57,188
CRT： ……166
CSNG ……188
CSRLIN ……190

CVD ···· 190
CVI ···· 190
CVS ···· 190

D

DATA ···· 57,176
DATE ···· 187,200
DEFDBL ···· 176
DEFFN ···· 176
DEFINT ···· 176
DEFSNG ···· 176
DEFSTR ···· 176
DEFUSR ···· 176
DEL ···· 200
DELETE ···· 33,176
DIM ···· 59,176
DIR ···· 143,201
DISKCOPY ···· 141,201
Disk offline ···· 207
DRAW ···· 178
DSKF ···· 192

E

ECHO ···· 201
ELSE ···· 53,177
END ···· 177
EOF ···· 191
ERA ···· 201
ERASE ···· 201
ERL ···· 188
ERR ···· 188
ERROR ···· 187
EXIT ···· 201
EXP ···· 188

F

FIELD ···· 180
FILES ···· 40,182
FIX ···· 188
FIXDISK ···· 201
FOR ···· 27,177
FORMAT ···· 201
FRE ···· 191

G

GET ···· 180,187
GOSUB ···· 172,177
GOTO ···· 48,177
GRP: ···· 166

H

HELP ···· 201
HEX$ ···· 190

I

IF ···· 53,177
IMP ···· 193
INKEY$ ···· 173,191
INP ···· 192
INPUT ···· 26,177
INPUT# ···· 168,180
INPUT$ ···· 191
Input past end ···· 207
INSTR ···· 190
INT ···· 51,188
INTERVAL ···· 173,182

K

KEY ···· 176,178,182
KILL ···· 41,182
KMODE ···· 202

L

LEFT$ ···· 189
LEN ···· 190
LFILES ···· 41,182
LINE ···· 67,178
LINE INPUT# ···· 180
LIST ···· 31,176
LLIST ···· 176,183
LOAD ···· 39,180
LOC ···· 191
LOCATE ···· 47,178
LOF ···· 191

LOG ··· 188
LPOS ··· 191
LPRINT ··· 183
LPT: ··· 166
LSET ··· 180

M

MAXFILES ··· 170,181
MEM: ··· 166
MERGE ··· 181
MD ··· 202
MID$ ··· 177,189
MKD$ ··· 190
MKI$ ··· 190
MKS$ ··· 190
MOD ··· 52,193
MODE ··· 202
MOVE ··· 202
MVDIR ··· 202

N

NAME ··· 41,182
NEW ··· 28,176
NEXT ··· 27,177
NOT ··· 193

O

OCT$ ··· 190
OFF ··· 173,182,183
ON ··· 173,182,183
ON ERROR GOTO ··· 173,187
ON n GOSUB ··· 172,177
ON n GOTO ··· 82,177
ON INTERVAL GOSUB ·· 183
ON KEY GOSUB ··· 173,183
ON SPRITE GOSUB ·· 173,183
ON STOP GOSUB ··· 173,183
ON STRIG GOSUB ·· 173,183
OPEN ··· 168,181
OR ··· 52,193
OUT ··· 183

P

PAD ··· 192
PAINT ··· 67,178
PATH ··· 202
PAUSE ··· 203
PLAY ··· 85,90,94,185
POINT ··· 190
POKE ··· 183
PRESET ··· 67,178
PRINT ··· 178,179
PRINT# ··· 168,181
PSET ··· 72,179
PUT ··· 169,181
PUT KANJI ··· 72,179
PUT SPRITE ··· 80,179

R

RAMDISK ··· 171,203
RD ··· 203
READ ··· 57,177
REM ··· 47,176,203
REN ··· 203
RENAME ··· 203
RENUM ··· 33,176
RESTORE ··· 177
RESUME ··· 173,187
RETURN ··· 172,177
RIGHT$ ··· 189
RMDIR ··· 203
RND ··· 51,188
RNDIR ··· 203
RSET ··· 181
RUN ··· 30,41,177

S

SAVE ··· 38,181
SCREEN ··· 64,179
SET ··· 204
SET ADJUST ··· 20,187
SET BEEP ··· 185
SET PAGE ··· 180
SET PASSWORD ··· 187

SET PROMPT ………… 187
SET SCREEN ………… 187
SET SCROLL ………… 163,180
SET TITLE ………… 187
SGN ………… 188
SIN ………… 57,188
SOUND ………… 185
SPACE$ ………… 189
SPC ………… 189
SPRITE ………… 173,183
SPRITE$ ………… 79,193
SQR ………… 57,188
STEP ………… 178,179
STICK ………… 81,192
STOP ………… 177,182,183
STR$ ………… 190
STRIG ………… 183,192
STRING$ ………… 189
SWAP ………… 187
Syntax error ………… 17,208

T

TAB ………… 189
TAN ………… 188
THEN ………… 53,177
TIME ………… 193,204
TROFF ………… 177
TRON ………… 177
TYPE ………… 204

U

UNDEL ………… 204
USING ………… 179,183
USR ………… 192

V

VAL ………… 80,190
VARPTR ………… 191
VDP ………… 193
VER ………… 204
VERIFY ………… 205
Verify error ………… 208
VOL ………… 205
VPEEK ………… 191
VPOKE ………… 183

W

WAIT ………… 183
WIDTH ………… 180

X

XCOPY ………… 205
XDIR ………… 206
XOR ………… 193

資料編

推奨図書一覧

ＭＳＸホームコンピュータ読本	（アスキー）
新・ＭＳＸの基礎知識	（アスキー）
ＭＳＸ２パーソナルユースのすべて	（アスキー）
ＭＳＸ２ ＢＡＳＩＣ 入門	（アスキー）
ＭＳＸビギナーズＢＡＳＩＣ	（アスキー）
ＭＳＸビギナーズ・ハンドブック	（エム・アイ・エー）
ＢＡＳＩＣのコツ	（アスキー）
エラー撃退ミニ事典	（アスキー）
ＢＡＳＩＣゲーム教室	（アスキー）
ＭＳＸ ＢＡＳＩＣゲーム集①	（エム・アイ・エー）
ＭＳＸ ＢＡＳＩＣゲーム集②	（エム・アイ・エー）
ＭＳＸ ＢＡＳＩＣゲーム集③	（エム・アイ・エー）
プログラム Ｄ．Ｊ．	（アスキー）
面白パズルブック	（アスキー）
アニメ Ｃ．Ｇ．に挑戦	（アスキー）
マイコン・ジュークボックス	（アスキー）
トランプゲーム集	（アスキー）
音楽のおくりもの	（アスキー）
ことばの実験室	（アスキー）
アクションゲーム３８	（アスキー）
知能ゲーム３８	（アスキー）
ＭＳＸ２テクニカルハンドブック	（アスキー）
ＭＳＸ２＋パワフル活用法	（アスキー）

漢字コード表の補足説明

パソコンの取扱説明書や内蔵ソフト使用説明書などの「ＪＩＳ第１水準漢字コード表」は、主に内蔵ソフトや付属ソフトで使用するよう作られているため、一部の記号などのコードがＢＡＳＩＣとは異なります。
ＢＡＳＩＣでＪＩＳ非漢字（ひらがな、カタカナ、英数字、ギリシャ文字、ロシア文字、記号など）を使用するときは、下の表を参照してください。

ＪＩＳコードは縦軸の値（Ａ）と横軸の（Ｂ）との和となります。
例）●のコード＝2160＋1C＝217C　　（数値はいずれも16進数）

	A＼B	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20
記号	2120		、	。	，	．	・	：	；	？	！	゛	゜	´	｀	¨	＾	￣	＿	ヽ	ヾ	ゝ	ゞ	〃	仝	々	〆	〇	ー	―	‐	／	＼
	2140	～	∥	｜	…	‥	‘	’	“	”	（	）	〔	〕	［	］	｛	｝	〈	〉	《	》	「	」	『	』	【	】	＋	－	±	×	÷
	2160	＝	≠	＜	＞	≦	≧	∞	∴	♂	♀	°	′	″	℃	￥	＄	¢	£	％	＃	＆	＊	＠	§	☆	★	○	●	◎	◇		
	2220	◆	□	■	△	▲	▽	▼	※	〒	→	←	↑	↓	〓	㈱	㈲	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	g	m	cm	mm	km	cm²
	2240	m²	ml	mg	kg	us	㈹	KK	TEL	No.	インチ		グラム	メートル	センチ	ミリ	ヘクタール	アール	ドル	トン	キロ	キログラム	キロワット	ワット	リットル	パーセント	ページ	ヤード	フィート	①	②	③	④
	2260	⑤	⑥	⑦	⑧	⑨	⑩	【	】	〠	☎	□	∫	√	‾	⊓	‖	⊔	⌒	◡	［	＿	］	（	）	(CR1)	(CR2)	(禁)	(均)	(↑)	(↓)		
英数字	2320	◀	◀	‥	▪▪	▪	：	⇨									０	１	２	３	４	５	６	７	８	９							
	2340	Ａ	Ｂ	Ｃ	Ｄ	Ｅ	Ｆ	Ｇ	Ｈ	Ｉ	Ｊ	Ｋ	Ｌ	Ｍ	Ｎ	Ｏ	Ｐ	Ｑ	Ｒ	Ｓ	Ｔ	Ｕ	Ｖ	Ｗ	Ｘ	Ｙ	Ｚ						
	2360	ａ	ｂ	ｃ	ｄ	ｅ	ｆ	ｇ	ｈ	ｉ	ｊ	ｋ	ｌ	ｍ	ｎ	ｏ	ｐ	ｑ	ｒ	ｓ	ｔ	ｕ	ｖ	ｗ	ｘ	ｙ	ｚ						
ひらがな	2420	ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た	だ
	2440	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み	む
	2460	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ	ゐ	ゑ	を	ん													
カタカナ	2520	ァ	ア	ィ	イ	ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ	ダ
	2540	チ	ヂ	ッ	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ	パ	ヒ	ビ	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ	ム
	2560	メ	モ	ャ	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ	ヰ	ヱ	ヲ	ン	ヴ	ヵ	ヶ										
ギリシャ文字	2620	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω								
	2640	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ	ω								
ロシア文字	2720	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю
	2740	Я																а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н	о
	2760	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я															

松下電器産業株式会社　ワープロ事業部

〒571 大阪府門真市大字門真1006 電話(06)908-1151

Printed in Japan

DFQF2324ZB　SS0990-2100